Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/SE05/000222

International filing date: 18 February 2005 (18.02.2005)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/545,924
Filing date: 20 February 2004 (20.02.2004)

Date of receipt at the International Bureau: 11 May 2005 (11.05.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

PCT / SE 2005 / 0 0 0 2 2 2

2 0 -04- 2005

PA 1288270

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

March 21, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: ***60/545,924***
FILING DATE: ***February 20, 2004***

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

M. Sias

M. SIAS
Certifying Officer

13281 U.S.PTO
022004

Please type a plus sign (+) inside this box → [+]

Revised PTO/SB/16 (8-00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
Attorney Docket No. 43315-9468US

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

22154 U.S. PTO
60/545924
022004

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Gorme	Sande	Slependen, Norway
Jan	Bugge	Sandvika, Norway

☒ *Additional inventors are being named on the* 1 *separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

Method, system and virtual asset register

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☒ Customer Number [26694] → *26694* 26694 PATENT TRADEMARK OFFICE

OR *Type Customer Number here*

☒ Firm *or* Individual Name	VENABLE LLP				
Address	P.O. Box 34385				
Address					
City	Washington	State	DC	ZIP	20043-9998
Country	U.S.A.	Telephone	202.344.4000	Fax	202.344.8300

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification *Number of Pages* [28]

☐ CD(s), Number [ ]

☒ Drawing(s) *Number of Sheets* [19]

☒ Other (specify) Integrated Network Asset Management, NAM Workshop, 30 pages; and Architecture and Design Document, Utility System Integration-Phase 2: Data access and consistency, 34 pages

☒ Application Data Sheet. See 37 CFR 1.76

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check or money order is enclosed to cover the filing fees

☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: [22-0261]

☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($) [160.00]

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: ______.

Respectfully submitted,
SIGNATURE ________ Date [02/20/04]

TYPED or PRINTED NAME Eric J. Franklin

REGISTRATION NO. *(if appropriate)* [37,134]

TELEPHONE 202 344-4000

Docket Number: [43315-9468US]

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

VENABLE ATTORNEYS AT LAW

PROVISIONAL APPLICATION COVER SHEET

Additional Page

Docket Number	43315-9468US	Type a plus sign (+) inside this box →	+

INVENTOR(S)/APPLICANT(S)

Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
Svein	Vatland	Sandefjord, Norway
Tormod	Lund	Oslo, Norway
Erik	Svensson	Zürich, Switzerland
Lars	Gundersen	Oslo, Norway
Danny	Julian	Willow Spring, North Carolina
Claus	Vetter	Baden-Daettwil, Switzerland
Thomas	Werner	Baden-Daettwil, Switzerland

Number 2 of 2

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

THIS IS A PROVISIONAL APPLICATION.

INVENTORS

	GORM SANDE
Address	Juterudåsen 42A NO-1341 Slependen Norway
NATIONALITY	NORWEGIAN
	JAN BUGGE Måneveien 9 NO-1300 Sandvika Norway
NATIONALITY	NORWEGIAN
	SVEIN VATLAND
Address	Bugårdsgata 13, NO-3208 Sandefjord Norway
NATIONALITY	NORWEGIAN
	TORMOD LUND
Address	Edvard Griegs alle 7 NO-0479 Oslo Norway
NATIONALITY	NORWEGIAN
	ERIK SVENSSON
Address	xxxxxx xxxxxxx xxxxxxx
NATIONALITY	SWEDISH
	LARS GUNDERSEN
Address	Nonnegt. 17 NO-0656 Oslo Norway
NATIONALITY	NORWEGIAN
	DANNY JULIAN
Address	7021 Landingham Drive Willow Spring, NC 27592
NATIONALITY	US
	CLAUS VETTER
Address	Im Ergel 10, PF 10, 5405 Baden-Daettwil, Switzerland
NATIONALITY	GERMAN
	THOMAS WERNER
Address	xxxxxxx xxxxxxx, Switzerland
NATIONALITY	GERMAN
INVENTION TITLE	**Method, system and virtual asset register**
AGENT	Venables
AGENT DOCKET NUMBER	
ASSIGNEE	ABB Technology Ltd, Affolternstrasse 44 CH-8050 Zürich Switzerland
ASSIGNEE REFERENCE NUMBER	9648US/AT

9648US/AT
NW
20-02-2004 This is a provisional application

Method, computer based-system and virtual asset register

TECHNICAL FIELD.

The present invention is concerned with a method and computer-based system for controlling, monitoring and/or maintaining equipment in an electrical power distribution system. In particular it is concerned with a database consistency method and computer-based system that enables consistent retrieval, synchronisation and storage of data between a plurality of databases containing information and data relative to operating an electrical power network.

TECHNICAL BACKGROUND

Electrical power distribution network systems for industrial and residential power users typically comprise many and various types of distribution equipment located over a large geographic area. Most utilities operating a geographically distributed asset such as an electrical power network need a suit of IT support systems to manage the operation and maintenance of the assets. Today there is no common way of information integration between these systems which makes information retrieval difficult when more than one system is involved. For example, a power transformer device must be known in several systems. The customer information system has knowledge what customers are connected to the transformer. The Network Information system (NIS or GIS) has information about the geographical location of the transformer. The ERP system (Enterprise Resource Planning) has the maintenance history of the transformer and the SCADA system (Supervisory Control and Data Acquisition) knows the actual performance measurements, temperature, voltages and so on of the transformer.

A successful integration of a GIS (Geographical Information System) system, preferably with other systems such as a real time SCADA system with an ERP system would make power network information available to the ERP system. US 6,564,201 B1, entitled Expert designer system virtual plug-in interface; describes virtual plug-in interface for an expert designer system for use with one or more database system types operates. These systems include a geographic information system (GIS) and a work management system (WMS). It is described that the virtual GIS plug-in interface interacts with the GIS such that the expert designer system core functions independently of the type of GIS database system. The document discloses use of object-oriented programming architectures to make a virtual interface for GIS and/or WMS systems which is independent of specific database type or manufacturer and simplifies the task of designing GIS databases.

Integration of ERP and GIS database systems would facilitate automatic creation of work orders from the ERP system for condition-based maintenance dependent on more automatic or manually controlled outputs from, for example, a SCADA system. For example maintenance could be scheduled on a basis of accumulated short circuit limits for a given breaker. Another example would be automatic generation of work orders for inspecting a protection device used with low frequency a distribution or a feeder line. It would also be possible to validate that future scheduled maintenance activities is permissible with respect to other maintenance activities, switching status, available power production resources, transmission capabilities and forecasted consumption and so on.

One complication to be dealt with is that changes in the power grid assets would need to be reflected in the IT systems. In addition, several types of power devices need to be modelled in more than one IT system. For example, a power transformer device must be known in several systems an example previously

mentioned. For example, the customer information system (CIS) has knowledge of which customers are connected to the transformer, the NIS system has information about the geographical location of the transformer, the ERP system has the maintenance history of the transformer and the SCADA system has the real time and stored measurements taken at the transformer.

Within the power industry and network management industry a common approach to document exchange and conversion, CIM, Common Information Model, has been developed around the use of XML-based formats. More information on current practices and method for use of CIM/XML (Common Information Model/eXtensible Markup Language) for data exchange within the electrical power industry may be obtained from North American Electricity Reliability Council (NERC), Federal Energy Regulatory Commission (FERC). The CIM/XML standards greatly facilitate the exchange and automatic conversion of documents produced by one supplier of a part of the network or an equipment for the network so that a second supplier can receive, handle and re-use the technical data from the original documents without manual intervention, editing or re-inputting.

However there is a series of difficult challenges to be overcome to achieve the kind of integration desired for the separate IT systems described. A demand facing utility network owners and operators is to extract more value from the existing assets in a network utility, in terms of higher output without causing increased maintenance work, breakdowns or equipment loss. Another demand is to be able to integrate IT systems so as to make information accessible to all users who have an interest in the network. Manual linking and connections have been made in the past to exchange data between different IT systems, and to reconcile data for consistency. However this has been done manually or on a batch basis and has not been practically implemented on a real-time basis. The task of integrating separate IT systems is complicated. In particular, there is a

difficult technical problem of sharing data between different databases and at the same time achieving and maintaining data consistency between multiple IT systems. In addition, the real-time nature of power network operation demands of that data retrieval and/or communication can work at high speeds in a network utility and in an automated and effective way.

SUMMARY OF THE INVENTION

The present invention aims to solve one or more of the above problems.

According to one aspect of the invention, the object is achieved by the initially defined method.

According to another aspect of the invention, the aims are achieved by a software architecture including an consistency and mapping layer based on a structured text standard.

According to another aspect of the invention, the aims are achieved by register of power network held by a computer-based system.

According to another aspect of the invention, the aims are achieved by a computer-based system.

A major advantage of the present invention is that integration is carried out in such a way that data across the different systems is kept consistent. In addition, the invention provides as well a new and better platform with which to support asset management applications.

The system integration achieved by the invention provides interface and access advantages for users of power network systems such as:
a series of User Interface navigation displays used and operated by users with standard object-oriented navigation, selection,

input and display methods. The displays give timely access to all relevant information, and from all integrated IT systems. The integrated systems comprise data and data representations that are context sensitive, and provide simple and unified ways to navigate between different functional views, technical views or contextual views of the same process equipment, device, installation or other network asset. In addition the invention provides advantageously for one consistent asset representation using the Virtual Asset Register (VAR), with single data entry, automatic synchronization, data exchange between applications and both mapping to CIM/XML model as well as import & export to CIM/XML. The integrated systems do not require any other special provisions and may, for example, use applications that can work on one or more generic CIM models. The all-important consistency checks may take place in the background, and be thus not visible or apparent to a user.

The integrated systems provided by the invention have the advantage of providing means for:
-reduced data maintenance cost for the network,
-optimised service life of equipment,
-increased quality of asset data, and
-improved decision support.

As much of the invention is implemented in software and may be implemented by means of novel software architectures the necessary time and capital cost of including the invention in both new installations and existing installations is relatively low and therefore also very advantageous for that reason.

According to another aspect of the invention, the aims are achieved by (one or more) a computer program directly loadable into the internal memory of a computer or processor, comprising software code portions for performing the steps of the method(s) according to the invention, when said program is run on a computer or processor. The computer program is provided either

on a computer readable medium or through a network, a high-speed private network, such as a local area network or a wide area network including the Internet.

According to still another aspect of the invention, the objects are achieved by a computer-readable medium having at least one program recorded thereon, where the program is to make a computer or processor perform the steps of the method according to the invention, when said program is run on a computer or processor.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will now be described, by way of example only, with particular reference to the accompanying drawings in the attached presentation file which:

Figure 1 shows a schematic block diagram of different IT systems and databases integrated by a CIM/XML layer and an HMI layer according to an embodiment of the invention.

Figure 2 shows a schematic block diagram of integrated and different IT systems and databases in which the CIM/XML layer comprises a Virtual Asset Register and a data exchange middleware and supports an asset optimization application according to another embodiment of the invention.

Figure 3 is a table or matrix over functions, purposes and implementations for data exchange and other applications in a power network according to an embodiment of the invention.

Figure 4 shows a schematic block diagram of both predictable and condition based maintenance in provided by data exchange and other applications such as a CMMS (Computerised maintenance management system) in a power network.

Figures 5, 6, 7 show stages in a method for handling alarms and faults using an HMI to interface both a SCADA system and a maintenance or CMMS application in connection with data exchange and other applications in a power network according to another embodiment of the invention.

Figures 8, 9 show stages of a method for handling alarms and faults using an HMI to interface objects in both a SCADA system and a maintenance or CMMS application to GIS system data for the same object(s) in connection with data exchange and other applications in a power network according to another embodiment of the invention.

Figures 10, 11 show representations of steps in alarm handling using an HMI to interface objects in SCADA, maintenance or CMMS, GIS wherein information about an object, a breaker, from a maintenance or technical information database is simultaneously accessed, according to another embodiment of the invention.

Figure 12 is a schematic block diagram of different local IT systems and databases integrated by a CIM/XML layer using XLST transforms to map global objects to local objects consistently according to an embodiment of the invention.

Figure 13 is a schematic block diagram of different IT systems and databases integrated by a CIM/XML layer using adapters to map the XML, attribute changes and object according to an embodiment of the invention.

Figures 14, 15, 16 show schematic overviews for data consistency between different local IT systems and databases SCAD, ERP and GIS integrated by a CIM/XML layer, and with respect to mapping and attribute conflicts.

Figure 17 shows a schematic overview including data from a GIS system in respect of a new object added to the network;

Figure 18 shows schematically how that same new object may also be recognized in the SCADA and ERP systems, according to an embodiment of the invention; and

Figure 19 shows a flowchart for a method according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE EMBODIMENTS

Figure 1 shows schematically three separate systems which are each operating in a part of a power network. It shows a SCADA (Supervisory Control and Data Acquisition) system 2a, with a database 2b; a GIS (Geographical Information System) system 3a with a database 2b, and a CMMS (computerised maintenance management system) 4a and database 4b. A wall 7 is shown symbolically separating each system. A first layer 1 is shown bridging the otherwise separate IT systems. Layer 1 is a CIM/XML layer for data consistency and/or synchronisation. A user interface navigation layer 5 which, as will be described in more detail later, also comprises one or more HMIs and is also shown bridging the otherwise separate systems.

Figure 2 shows schematically an architecture in more detail. It shows separate IT systems for operations of the power network such as SCADA, EMS (Energy Management System)2a and DMS (Distribution Management System) 2b; a maintenance system such as a CMMS 4a, b; and GIS or NIS systems 3,a, b. The IT systems are connected via a middleware EAI (Engineering Application Integration) layer 14 and a Virtual Asset Register (VAR) 10 to a user interface navigation layer 5' that comprises one or more HMIs. Also included in the architecture are one or more applications for asset management 12.

Figure 3 shows in a matrix form various network operation functionalities, under the heading Customer value, with a

comment or expanded explanation under the heading Comment, and possible implementations under the heading Mapping. It shows in Step 1 that the Navigation functions of the present invention are context sensitive, depending on which context i.e. which system is currently activated by the user, which is achieved and implemented by object and object-oriented architecture to enable linking between the representations of the same object in all systems. Step 2 shows that Consistency means that objects may be added and or deleted in a consistent way across all systems, implemented by a grouping function, a hierarchical parent-child type of grouping model using object-oriented references also referred to as structures; and that data exchange is carried out consistently and based on a CIM model and implemented by means of checking attributes of the objects in the separate IT systems for consistency with stored values for such attributes.

Step 3, Asset Optimization is primarily concerned with optimizing the service life of equipment in the network, for example by means of an optimised balance between condition-based maintenance and/or predictable maintenance and/or planned maintenance. This may be carried out by the use of a CIM/XML model and mapping between systems, eg mapping from a given fault object reported by a SCADA system to the same given object held in a maintenance system such as a CMMS system, which mapping is carried out by means of the virtual asset register (VAR) as described in more detail below with reference to Figures 12, 13.

Figure 4 shows an optimised example for handling maintenance faults according to an embodiment of the invention. The figure shows an equipment 40 monitored by a SCADA system, and modeled on an HMI of a operation system 41. The figure shows that an operator 43 may provide operator input through an application 43c of the HMI which may be communicated to a Predictable Maintenance (PM) or Work Order (WO) application 45b of a PM or WO system. The figure also shows that the SCADA system collects data from field devices 43a, such as the above equipment 40,

and provides alarms and/or data events, parameters etc 48 to a PM system 44. Automation equipment can trigger work orders based on real time information from the equipment itself, for example operational hours, fieldbus information, maintenance triggers or software agent-type maintenance triggers. It can further be seen that as well as data collection from devices, direct input 43b from a control centre operator and/or direct input 45c from an engineer in the field with a portable device may be received by the PM system and be then available to the other systems. The Work Order (WO) system may be a part of or module for a maintenance system, or it may be a part of an ERP (Enterprise Resource Planning) system. Alternatively direct input 43b may be generated by an automatic, computerised process instead of or as well as a human operator.

Figure 5 shows in more detailed description a stage of handling a fault in an equipment similar to that fault described in relation to Figure 4. A fault is reported by the SCADA system, in Bay 5, 50 and the faulty object may be selected on a screen of an HMI of a Navigation System accessing the SCADA system data. The user is then presented with options, in this example by means of a drop-down window 51. Figure 6 shows how the user may select Active Work Orders 52 to check the current maintenance situation for this object, a breaker. An interface element, in this case the window 51, accessing Active Work Order status 53 retrieved from a maintenance system such as a CMMS (Computer managed maintenance system), shows the faulty equipment, the object, and any active work orders for it. Figure 7 shows how, using a fault reporting element 55 of the HMI interface, the user may file a fault report which may become the basis of a new work order. This also corresponds to the fault report 43c of Figure 4.

Figure 8 presents information about the same selected faulty object 40', the breaker, from the Geographical Information System (GIS), giving an overview of the geographical location of

the breaker. Figure 9 presents a view of the HMI with which a user may access a work order, in this case the new work order 63 for the breaker by means of options in a drop down window operable from the selected screen object from the location display provided by GIS information.

Figure 10 is similar to Figure 5, showing the faulty equipment with options displayed, with in this case, an option to access 59 the Maintenance and Service Manual for the selected object, the breaker, which produces product documentation 100. Figure 11 shows how the HMI in the Navigation Interface System integrates information and access for the same selected object so that the SCADA system information 50, the GIS system information 40′, the CMMS work order information 63 and the CMMS Technical Manual information 100 may presented for a user to see, access and/or manipulate at the same time. In addition, distance and/or route to a site may be seen simultaneously from the GIS information which facilitates determining which repair crew should be dispatched and what extra factors concerned with distance to site, time to site and/ or details of site topology need to be considered. The display shown by Figure 11 comprises then information from separate IT systems, SCADA, GIS, CMMS, that is to say, data and/or information accessed in the separate IT systems and retrieved at the same time and displayed together by the computer-based system for the user. The user has full access to data held by each of the IT systems by provided by the invention, which data is maintained in a fully consistent state by the consistency mechanisms of the invention.

Figure 19 shows the above method in the form of a flowchart. Some of the steps of the method are carried out by means of computer programs. The steps of the method in this exemplary example begin with a signal from one particular IT system of the network, in this case and not exclusively, beginning at 80 with a SCADA report, in this case and not exclusively in the practice of the invention, a fault report. At 81 the faulty object is

selected on a display by a user, operator. This step may alternatively be carried out by a process running in a computer, that, in effect, executes a process that has the equivalent effect of selecting the faulty object. At 82 a user checks maintenance information, for example to see what work orders are active in a CMMS system. The SCADA system is operated 84 to isolate an equipment and to restore the network to an operating condition; this may be done by an operator, a process in a computer, or by a combination. This means then that the SCADA system is operated, ie control signals are generated by the SCADA system so as to switch lines and/or equipment on or off, in this case, for the purpose of isolating an equipment or part of a line. At 85 a fault report may be created by the operator in the CMMS system 86, or semi-automatically, or automatically by CMMS. On the basis of the information provided in the computer-based system from SCADA, CMMS and GIS, an operator or a pre-programmed process may dispatch a work crew 87 to a fault location, such that the crew as well as other users of the computer-based system has access to the GIS geographical information, map, to find the best route to the location. At the same time, product documentation is retrieved 89 and accessed 90 so that detailed maintenance information for the equipment of interest is simultaneously available throughout the computer-based system for any validly logged on user.

Figure 12 shows in a schematic block diagram an overview of the global-local relationships, and that part of how data consistency is maintained. Figure 12 shows local databases, accessible in relation to local objects, linked to a set of global objects comprised in a virtual asset register (VAR). The figure shows a local SCADA database 2b, a GIS database 3b and a CMMS database 4b. It also shows a VAR 10 linked by adapters or in this case transform means XLST 2t, 3t, 4t to a local object, 2o, 3o, 4o, and thus by means of those two functional elements to each of the three local databases 2b, 3b, 4b. Global objects which are based on a CIM/XML model are

maintained in the VAR 10 as links to objects, not objects as such, only in the form of cross-reference and mapping data for each object in the power network.

The XLST transforms are a preferred adapter implementation for translating the XML based CIM model data into a format that can operated on local objects 2o, 3o, 4o which provide access and retrieval, read and write access, into the local SCADA, GIS and CMMS databases. This access is not necessarily identical and may well be different for different systems or databases. All participating applications must provide read and write access to their data sets through APIs (database access, OPC access, direct API access), where APIs are Application Programming Interfaces and OPC is an industry standard for linking or locating data called Object (Linking and Embedding) for Process Control.

The VAR 10 consists of the following components:
-A global CIM/XML based data model for exchange of data between the involved applications (typically GIS, CMMS and SCADA)
-A mechanism and means for Data Consistency. If an object is created or modified in one of the systems, the changes are reflected also in the other systems. This means that the object is automatically created or the attributes are changed in the other systems. The data consistency checking may be run as a real-time process or alternatively may be run as a batch job that is run once a day for example.
-a database, preferably an SQL (Structured Query Language) database, for example an Oracle (TM) or MS SQL Server (Microsoft SQL server TM) that contains cross-reference/mapping tables for objects in the different applications and tables of object types that describe what attributes the objects have in each application. This is especially important when each application has its own names for the same objects and attributes. It should be noted that this database contains only cross-reference and mapping data: in this sense it is a virtual asset register (VAR)

because it does not contain the actual object data as such. It contains only the cross-reference and mapping data and the actual data that describes each object is stored in each applications database, that is, in the SCADA system, the GIS system, the CMMS system and so on.

Figure 13 shows the arrangement of Figure 1 with the role of adapters, such as the adapters shown and described in relation to Figure 12. The figure shows a possible inclusion of applications known as Message oriented Middleware (MoM). It also shows that other database systems Na may optionally be included in the computer-based system as well as SCADA, GIS, CMMS systems. For example other related and/or legacy systems such as CIS, PM, WMS, WOS.

Figure 14 shows a display and input member for consistency checking between the separate IT systems. It shows a SCADA system 2, SAP (CMMS) system 4, and an ESRI (GIS/NIS) system 3. (SAP is a trademark). Figure 15 shows a schema for mapping between global objects and local objects which is part of the consistency checking functions described in relation to Figure 14. It can be seen in this example that CIM-type Global substation objects LoadArea, MemberOF, name and Description are in this case mapped to the following local substation objects GENERALPROPERTIES.LOADAREA, GENERALPROPERTIES.MEMEBEROF, NAME.NAME and NAME.DESCRIPTION respectively. Figure 16 shows conflict handling within the consistency functionality. Here it may be seen that one three IT sytems, AIP 117, ESRI 118 and SAP 119 are available fro a user or engineer to select as Master data or source data to resolve a potential consistency conflict.

Figure 17 shows the insertion of a new object 170 in a display from a GIS system. Figure 18 shows a combination of the GIS and a synchronization window 180, which may be a part of the data consistency functions. It may be seen that the GIS system ESRI has flagged a new object substation.xsd, and options are

presented to Insert in SCADA 189, or insert in SAP 187. The option to insert in ESRI 188is greyed, as the object has already been detected as inserted in the GIS system ESRI.

The integration may be implemented by means of a SCADA system with a SCADA user interface, of the WS500 type of the Spider system for SCADA provided by ABB, and a CMMS maintenance management system from IFS, a NIS or GIS system such as ESRI planning & mapping system and a HMI integration platform and/or application integration platform such as an ABB Industrial IT system from ABB. The invention demonstrates a seamless user interface integration between SCADA, CMMS or GIS with context sensitive access to CMMS from SCADA or GIS (by means of object linking) in a computer-based system comprising access to those separate IT systems described.

The invention makes it possible to operate one or more parts of one or more power networks as one global or enterprise level data model of the assets (CIM+).

Attribute consistency - updating attribute values (overlapping)
Object consistency - adding/deleting objects:
Single data entry
One consistent enterprise level : Virtual Asset Register (VAR)
Add new object (in all relevant systems)
Object created in each system based on object templates
Connections between systems established automatically
Delete object (in all relevant systems)
Delete defined object in each system
Delete object connections (links)
Access object attributes (all)
Select object by identifier (any system)
Read out any object property independent of source
Modify object attribute(s)
Select object by identifier (any system)
Update attribute in source system (owner)

Replicate data to other systems (readers of the data)
Maintain object connections (links)

Power network equipment in an Energy Management System (EMS) or Distribution Management System (DMS) may include any combination or combinations of transmission lines, distribution lines, transformers or reactors of various types, switchyards, substations, protection devices, live tank circuit breaker, disconnector, switch-disconnector or load disconnector, earthing switch, disconnector circuit breaker, dead tank circuit breaker, gas-insulated circuit breaker, gas-insulated disconnector, earthing switch, switchgear modules including CBs, DCs, SDs etc as above.

CIM, Common Information Model, is an industry standard approach in the Energy Management System industry covering the use of XML formats in data exchange. More information on current practices and method for use of CIM/XML for data exchange within the electrical power industry may be obtained from North American Electricity Reliability Council (NERC), Federal Energy Regulatory Commission (FERC).

The CIM language includes a set of class diagrams that use the UML, Unified Modeling Language. CIM/XML may be described as the incorporation of elements from the RDF (Resource Description Framework, as defined by W3C) data model to form CIM/XML. For example by using an RDF element, a URI (Uniform Resource Identifier), to represent resources. Resources may correspond to objects and properties may correspond to object attributes.

An XML grammar, as defined in a suitable DTD (Document Type Definition) can be used both to represent CIM declarations (classes, instances and qualifiers) and CIM messages for use by the CIM mapping onto another protocol such as HTTP. Mapping with an XML derivative may be carried out using any suitable approach, such as schema mapping in which the XML Schema is used

to describe the CIM classes, and CIM Instances are mapped to valid XML Documents for that schema; or meta-schema mapping in which the XML schema describes the CIM meta-schema, and both CIM classes and instances are valid XML documents for that schema.

Use of an XML or XML/CIM format may include the use of stylesheets and in particular XSLT stylesheets. A well-formed XML document may include both elements that are defined by XSLT and elements that are not defined by XSLT. XSLT stands for eXtensible Stylesheet Language Transform - thus it is a programming language, or other means, for transforming XML documents and rendering them in HTML or between different formats. XSLT-defined elements are distinguished by belonging to a specific XML namespace. A transformation expressed in XSLT is called a stylesheet. This is because, in the case when XSLT is transforming into the XSL formatting vocabulary, the transformation functions as a stylesheet.

Other current standards capable of use for data exchange include derived protocols such as COM (Component Object Model) Document Object Model (DOM), Microsoft's (Trade Mark) MSXML and a standard called XHTML 1.0 provided by World Wide Web Committee (W3C). The invention is not limited to XML based implementations and may alternatively use any derivative of a format such as the Standard Generalised Markup Language (SGML) meta-language, or Hyper Text Markup Language (HTML), eXtended Markup Language (XML) or derivatives such as XHTML 1.0, Extended Stylesheet Language (XSL) and the Document Object Model (DOM); or adaptions suited for user to handle using applications on portable or mobile devices, for example Wireless Markup Language (WML), which may be used with a WAP telephone may be described as a derivative of XML, or a WDML derivative, or WBXML.

The client applications of the HMI may be implemented as a thin client using a structured text document or file to present any

of CIM/XML information, arguments, variables, addresses, links, mappable objects, executable applications or applets, or for example an HTML or other WWW based or HTML derivative protocol or XML protocol. The structured text document or file format takes care of handling graphical user display and activation functions of the HMI client. Activation functions refers to functions in the web page or web client display carried out by executable applications or applets which may be implemented as Java (TM) or similar. By means of such a thin client version of the HMI with an architecture such as that shown in Figure 1, 2, 4, 19, a user or a technician may examine status or data, configure a parameter, change set points and/or issue commands remotely in to any object for which he/she has authority to so do via the navigation interface.

The methods of the invention may be carried out by means of one or more computer programs comprising computer program code or software portions running on a one or more servers, a computer, or a processor. The computer or microprocessor (or processors) comprises a central processing unit CPU performing the steps of the method according to one or more facets of the invention, such as the methods described. The methods are performed with the aid of one or more said computer programs, which are stored at least in part in memory accessible by the one or more processors.

For example a program or part-program that carries out some or all of the steps of methods such as that described in relation to Figure 19, may be run by a computer or processor of the computer-based system. At least one of the servers or computers may be in a central object oriented control system in a local or distributed computerised control system. It is to be understood that said computer programs may also be run, at least in part, on one or more general purpose industrial microprocessors or computers instead of one or more specially adapted computers or processors.

The computer program comprises computer program code elements or software code portions that make the computer perform the method using equations, algorithms, data, stored values and calculations previously described. A part of the program may be stored in a processor as above, but also in a ROM, RAM, PROM, EPROM, or EEPROM chip or similar memory means. The program in part or in whole may also be stored on, or in, other suitable computer readable medium such as a magnetic disk, CD-ROM or DVD disk, hard disk, magneto-optical memory storage means, in volatile memory, in flash memory, as firmware, stored on a data server or on one or more arrays of data servers, or in high security data storage systems. Other known and suitable media, including removable memory media such as removable flash memories, hard drives etc. may also be used at least in respect of part of the data.

Data may also be communicated wirelessly between various parts of a power network, and/or to or from one or more of the different IT systems and databases. For example data may be collected from sensors arranged on equipment on a line or in a switchyard or a substation, stored and communicated as necessary by a SCADA system. Wireless communications may be carried out using any suitable protocol, including a wireless telephone system such as GSM or GPRS. Signals from a SCADA, CMMS or other system may also be sent via wireless communication to an equipment in a power network arranged with wireless communication so as for example as a result of an update of maintenance status, to activate a control, to set a breaker to maintenance mode, or perform a control action. Short range radio communication is a preferred technology, using a protocol compatible with, standards issued by the Bluetooth Special Interest Group (SIG), any variation of IEEE-802.11, WiFi, Ultra Wide Band (UWB), ZigBee or IEEE-802.15.4, IEEE-802.13 or equivalent or similar. In particular a radio technology working in, for example, the ISM band with significant interference suppression means by spread spectrum technology is advantageous,

especially communication for field devices or sensors. For example a broad spectrum wireless protocol in which each or any data packet may be re-sent at other frequencies of a broad spectrum 7 times per millisecond, for example, may be used, such as in a protocol from ABB called Wireless interface for sensors and actuators (Wisa).

The computer programs described above may also be arranged in part as a distributed application capable of running on several different computers or computer systems at more or less the same time. Programs as well as data such as energy related information may each be made available for retrieval, delivery or, in the case of programs, execution over the Internet.

Data and/or methods may be accessed by software entities or other means of the control system by means of any of the lost of: OPC, OPC servers, an object request broker such as COM, DCOM or CORBA, a web service.

It is also noted that while the above describes exemplifying embodiments of the invention, there are several variations and modifications which may be made to the disclosed solution without departing from the scope of the present invention as defined in the appended claims.

CLAIMS

1. A method for retrieving and accessing data stored in a plurality of systems arranged for operating part of one or more electrical power networks which method comprises adding a new object into a first system, and subsequently adding a copy of the new object into all relevant systems, establishing automatically a connection between said relevant systems and the new object, and replicating data related to the new object to other systems and relevant systems.

2. A method according to claim 1, **characterised** by maintaining object connections (links) for the new object and for any other object accessed, retrieved and/or stored by a SCADA system as well as by any system from the list of: GIS system, ERP system, CMMS system, PM system, WO system, WMS system.

3. A method according to claim 2, **characterised** by establishing the consistency of accessed or retrieved data in the relevant systems by means of mapping the new object using a model based on a structured text document.

4. A method according to claim 3, **characterised** by checking the consistency of attributes of the accessed or retrieved data by identifying the new or a given object and/or copies of the new or a given object and comparing attributes of all copies of the same new or given object.

5. A method according to claim 4, **characterised** by mapping the new object and/or copies of the new object using a model based on a CIM/XML document.

6. A method according to claim 4, **characterised** by mapping attributes of the new object and/or copies of the new object using a model based on a CIM/XML document.

7. A method according to claim 1, **characterised** by establishing the automatic connection or connections between copies of the same object in different systems means of a CIM/XML layer (1).

8. A method according to claim 1, **characterised** by mapping the new object by means of a virtual asset register (10) dependent on the CIM/XML layer (1) and/or mapping.

9. A method according to claim 1, **characterised** by selecting an object by means of an identifier in any said relevant system.

10. A method according to claim 9, **characterised** in that the identifier may be a URI (Uniform Resource Identifier) compatible as an identifier with a standard for RDF (Resource Description Framework).

11. A method according to claim 6, **characterised** by accessing one or more object attributes of the new object and changing an object attribute of the new object in a source system (owner, the first system).

12. A method according to claim 6, **characterised** by updating an object attribute of the new object in the source system (owner, the first system).

13. A method according to claim 1, **characterised** by creating the new object in each relevant system based on object templates.

14. A method according to claim 1, **characterised** by deleting an object by deleting the object in all relevant systems.

15. A method according to claim 14, **characterised** by deleting an object by deleting a defined object in each system.

16. A method according to claim 15, **characterised** by deleting an object by deleting object connections (links) to a deleted object or deleted defined object.

17. A computer program for retrieving and accessing data stored in a plurality of systems plurality of systems arranged for operating part of one or more electrical power networks comprising software code portions or computer code to cause a computer or processor to carry out the steps of a method according any of claims 1-16.

18. A computer program product recorded on a computer readable medium which when read into a computer or processor will cause the computer or processor to carry out a method according the steps of claims 1-16.

19. An asset register for retrieving and accessing data stored in a plurality of systems arranged for operating part of one or more electrical power networks in which asset register is a list of power network assets which list comprises cross reference and mapping data for objects represented and/or stored in a SCADA system as well as in any system from the list of: GIS system, ERP system, CMMS system.

20. An asset register according to claim 19, **characterised** by a comprising a list of references for all objects representing individual items of physical and/or logical equipment comprised in the one or more parts of the said power network.

21. An asset register according to claim 20, **characterised** in that the list comprises a master list of all objects in the one or more parts of the said power network together with the mapping data for each object according to a CIM model.

22. An asset register according to claim 21, **characterised** in that object data for the objects comprised in the master list of

the asset register is stored in at least one separate system including any of a system for: SCADA, GIS, CMMS, ERP, PM, WO.

23. An asset register according to claim 21, **characterised** in that the asset register is a virtual asset register which does not comprise any object data for the objects comprised in the master list and only comprises link information or cross reference data or mapping data.

24. A human-machine interface for retrieving and accessing data stored in a plurality of systems arranged for operating part of one or more electrical power networks, which HMI comprises a display including data accessed or retrieved from or stored in a SCADA system, as well as data accessed or retrieved from or stored in any from the list of: GIS system, ERP system, CMMS system, PM system, WO system.

25. A human-machine interface according to claim 24, **characterised** by at least one graphical user interface with means for manipulation of the data retrieved from or stored in the SCADA and any of the systems for GIS and/or ERP and/or CMMS.

26. A human-machine interface according to claim 24, **characterised** by reading out any object property independent of source.

27. A human-machine interface according to claims 24-26, **characterised** by means to provide access to simultaneous data stored in or held by any of the list of: SCADA system, GIS system, ERP system, CMMS system, PM system, WO system.

28. A computer-based system for retrieving and accessing data stored in a plurality of systems arranged for operating part of one or more electrical power networks, which computer-based system comprises a plurality of databases and a data communication network and which system includes an HMI providing

navigation and access to at least one SCADA system and/or database as well as to any system and/or database from the list of: ERP, GIS, CMMS, WO, WMS, PM.

29. A computer-based system according to claim 28, **characterised** by comprising one or members for: adding a new object; automatically establishing a connection between said relevant systems and the new object; and for replicating data related to the new object to other systems and relevant systems.

30. A computer-based system according to claim 29, **characterised** by comprising one or members for: maintaining object connections; providing connection or connections by means of a layer with a structured text document protocol; and mapping the new object by means of a structured text document model.

31. A computer-based system according to claim 30, **characterised** in that any of: the structured text document protocol layer, or the structured text document means for mapping the new object are implemented with a CIM/XML model.

32. A computer-based system according to claim 28, **characterised** by comprising one or members for checking the consistency of attributes of any data so accessed or retrieved data by identifying the or each new or given object and/or copies of the new or given object in any separate system, and comparing attributes of all such copies of the same new or given object from each of the separate systems.

33. A computer-based system according to claim 28, **characterised** by a virtual asset register.

34. A computer-based system according to claim 28, **characterised** by a virtual asset register implemented according to an XML or CIM model or document.

35. A computer-based system according to claim 28, **characterised** by an HMI that may comprise object data accessed or retrieved or stored in a SCADA system and/or database as well object data originating in any system and/or database from the list of: ERP, GIS, CMMS, WO, PM.

ABSTRACT

The invention is a method and computer-based system for retrieving, accessing and storing data otherwise stored in a plurality of IT systems arranged for operating a part of one or more electrical power networks. The method and computer-based system provide integration of otherwise separate systems such as GIS, SCADA, ERP and/or CMMS. The integration includes an XML/CIM layer for data exchange and a virtual asset register containing references to objects representing assets in the power networks. In other aspects of the invention a method, a human-machine interface and a computer program for carrying out the method are described.

(Figure 1)

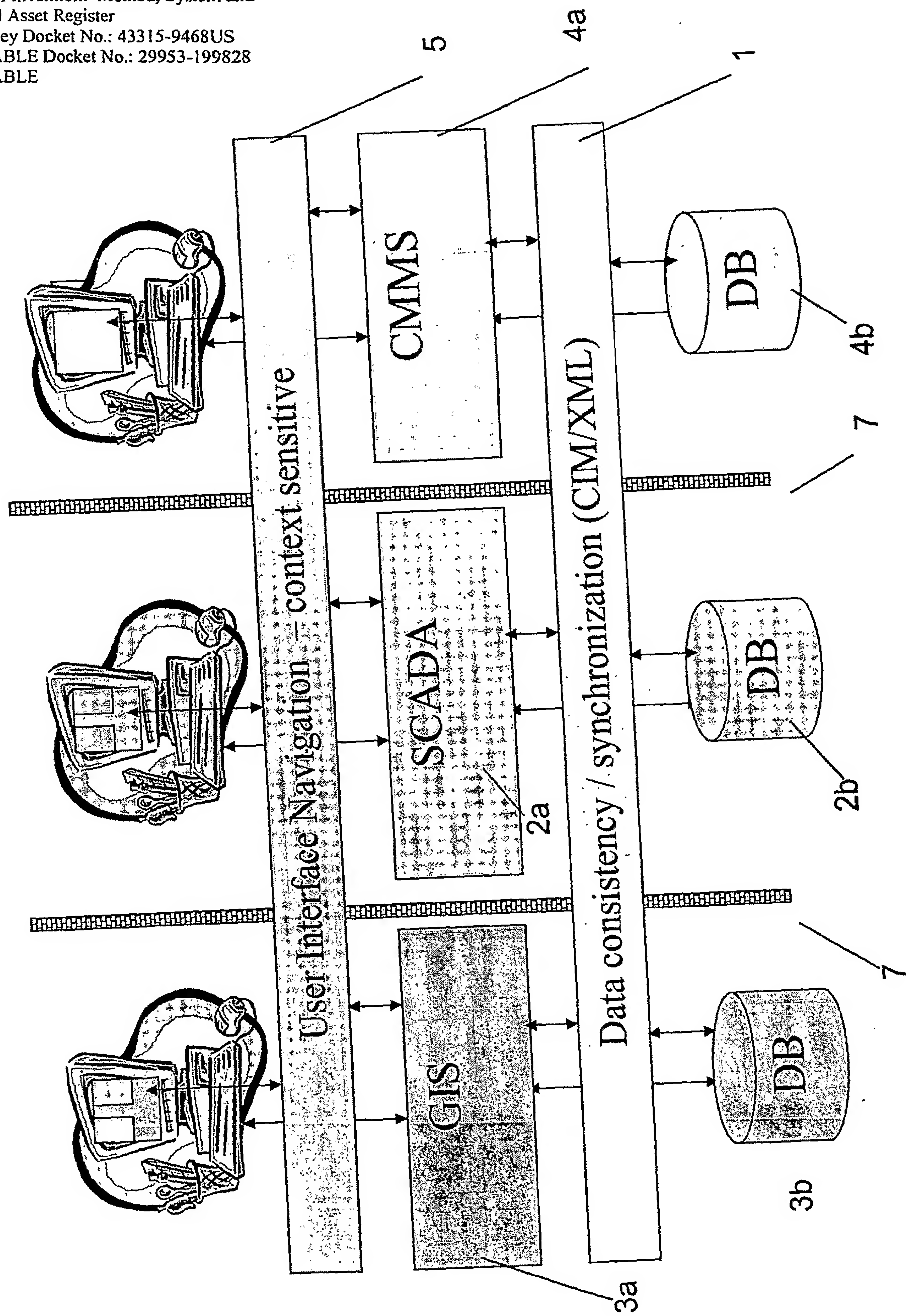

5
4a
1
User Interface Navigation – context sensitive
CMMS
SCADA
GIS
Data consistency / synchronization (CIM/XML)
DB
DB
DB
4b
7
2a
2b
7
3a
3b

Figure 1

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

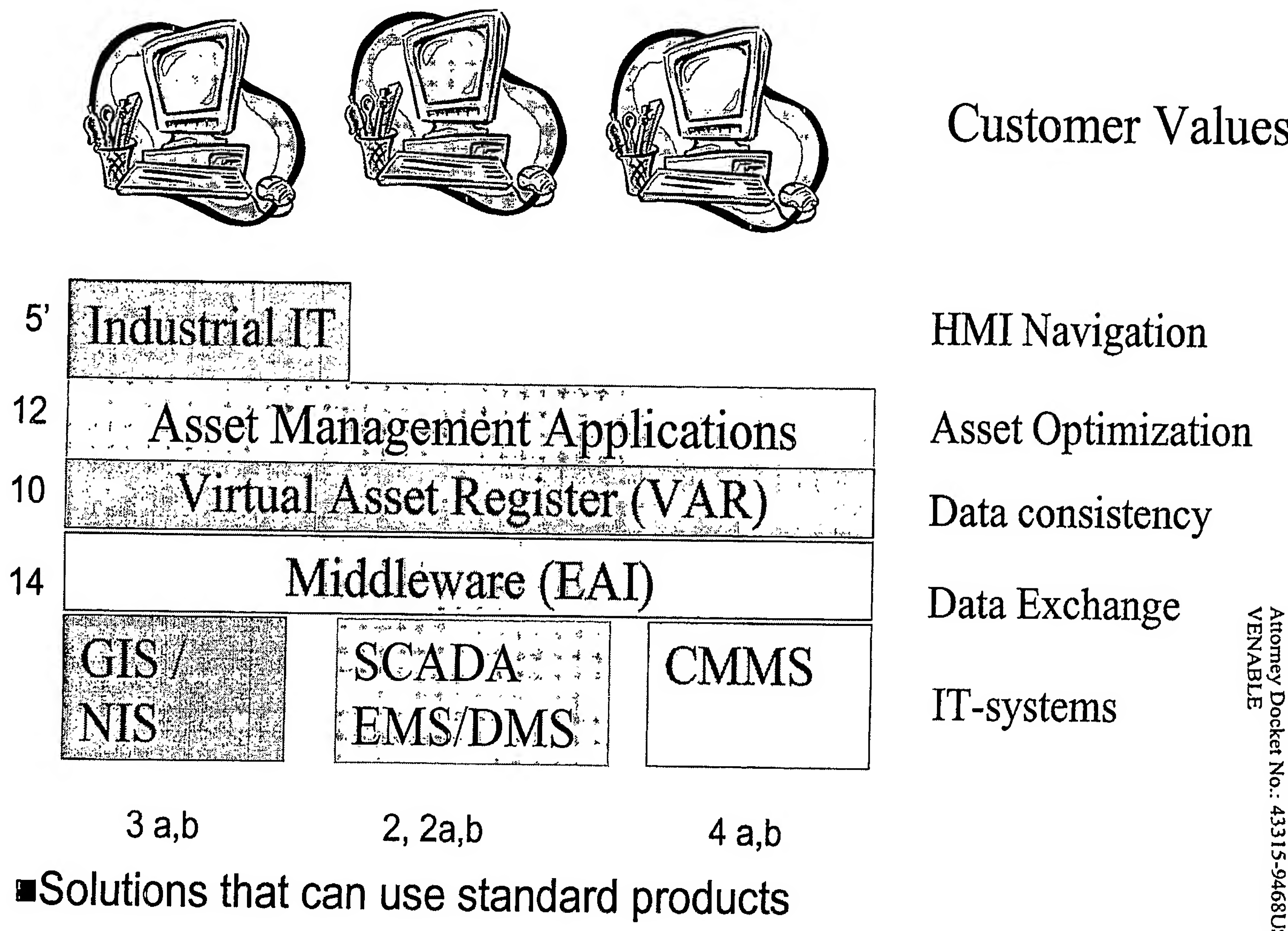


Figure 2

Development stage	Customer value	Comment	Mapping
Step 3	Asset optimization	Lifecycle, utilization	CIM/XML
Step 2	Consistency	Add/delete objects consistently	Structure
	Data exchange	Consistent values, CIM model	Attribute
Step 1	Navigation	Context sensitive, IIT (Aspect Object)	Object

Figure 3

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

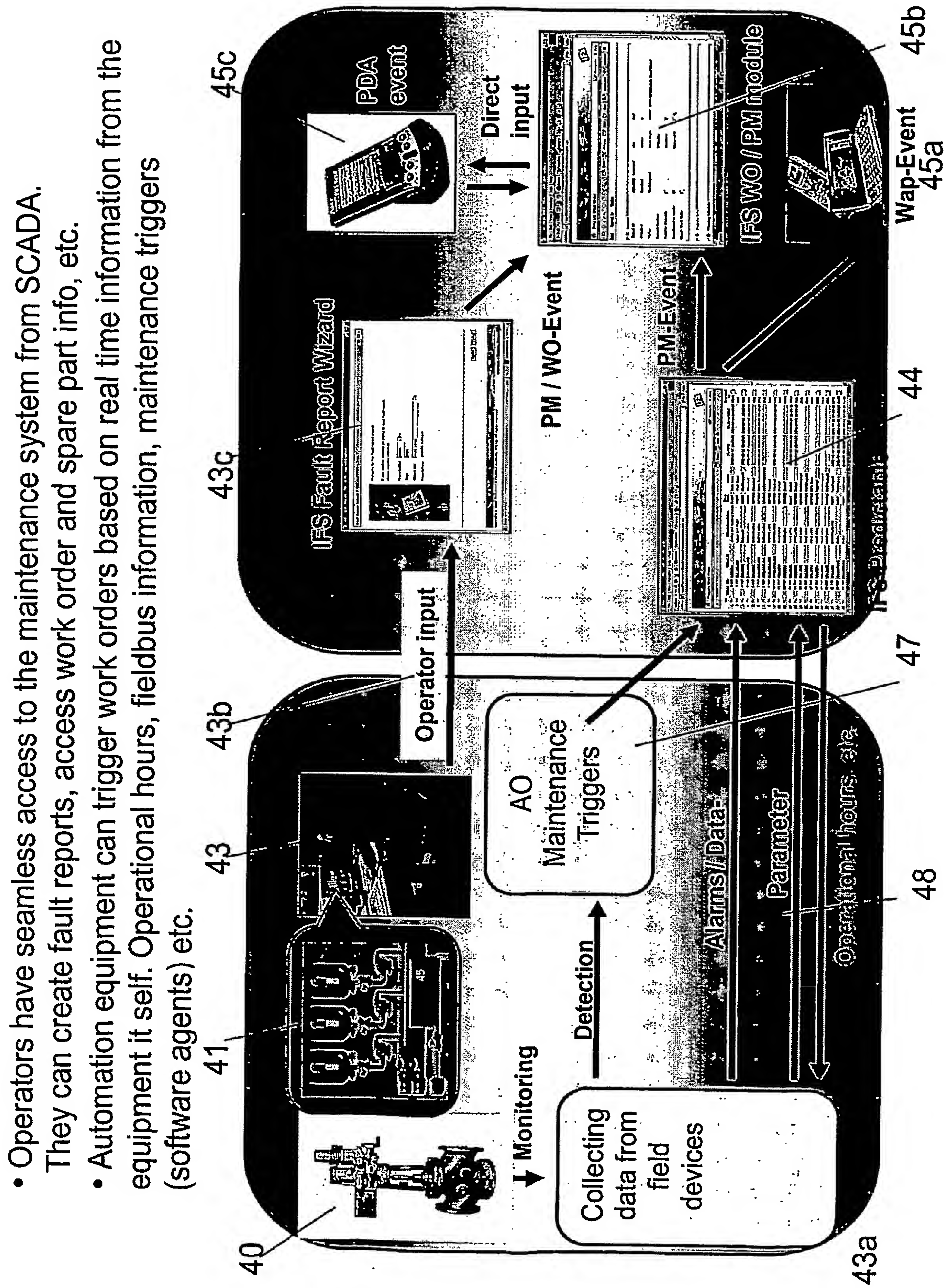


Figure 4

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

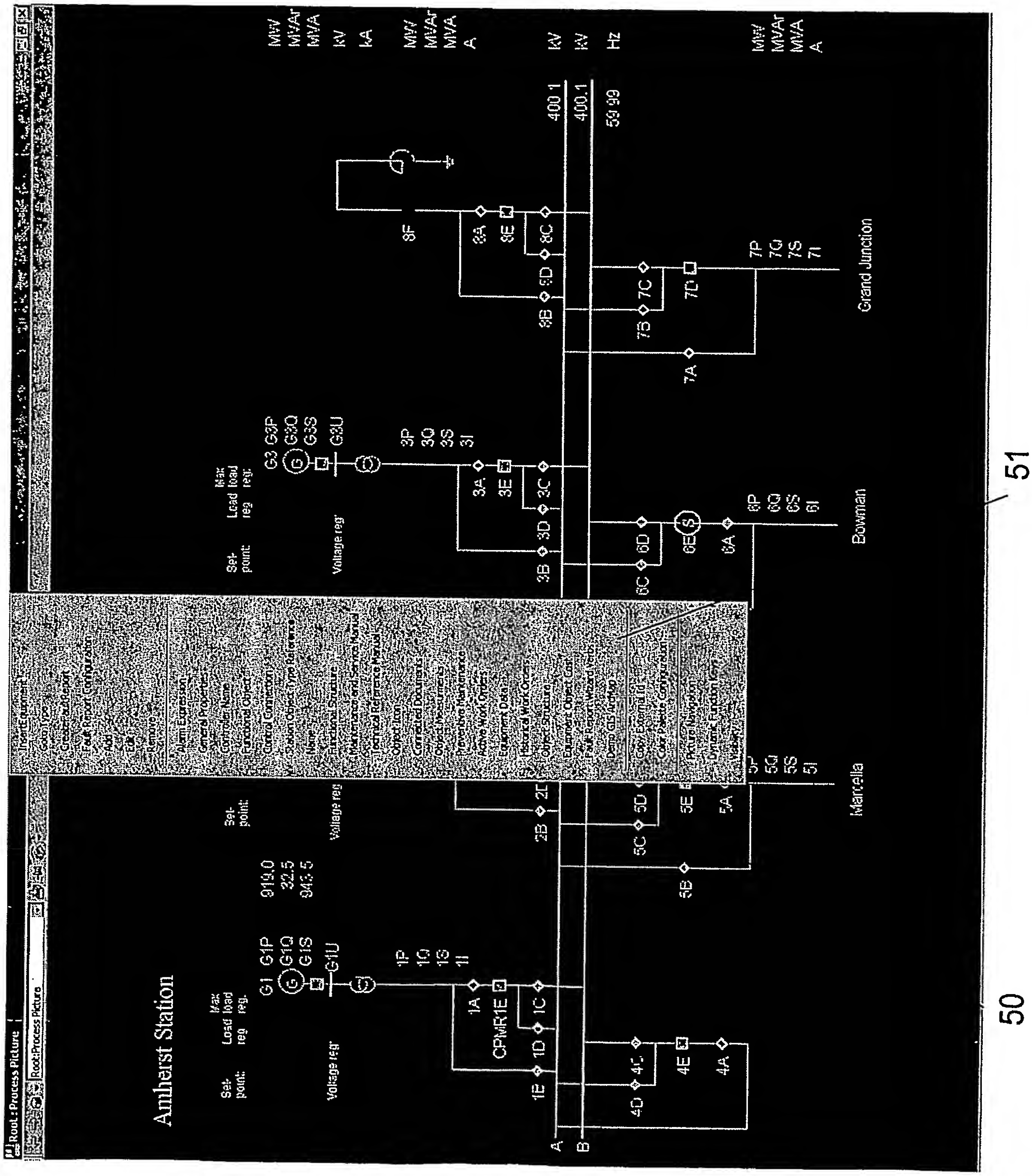


Figure 5

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

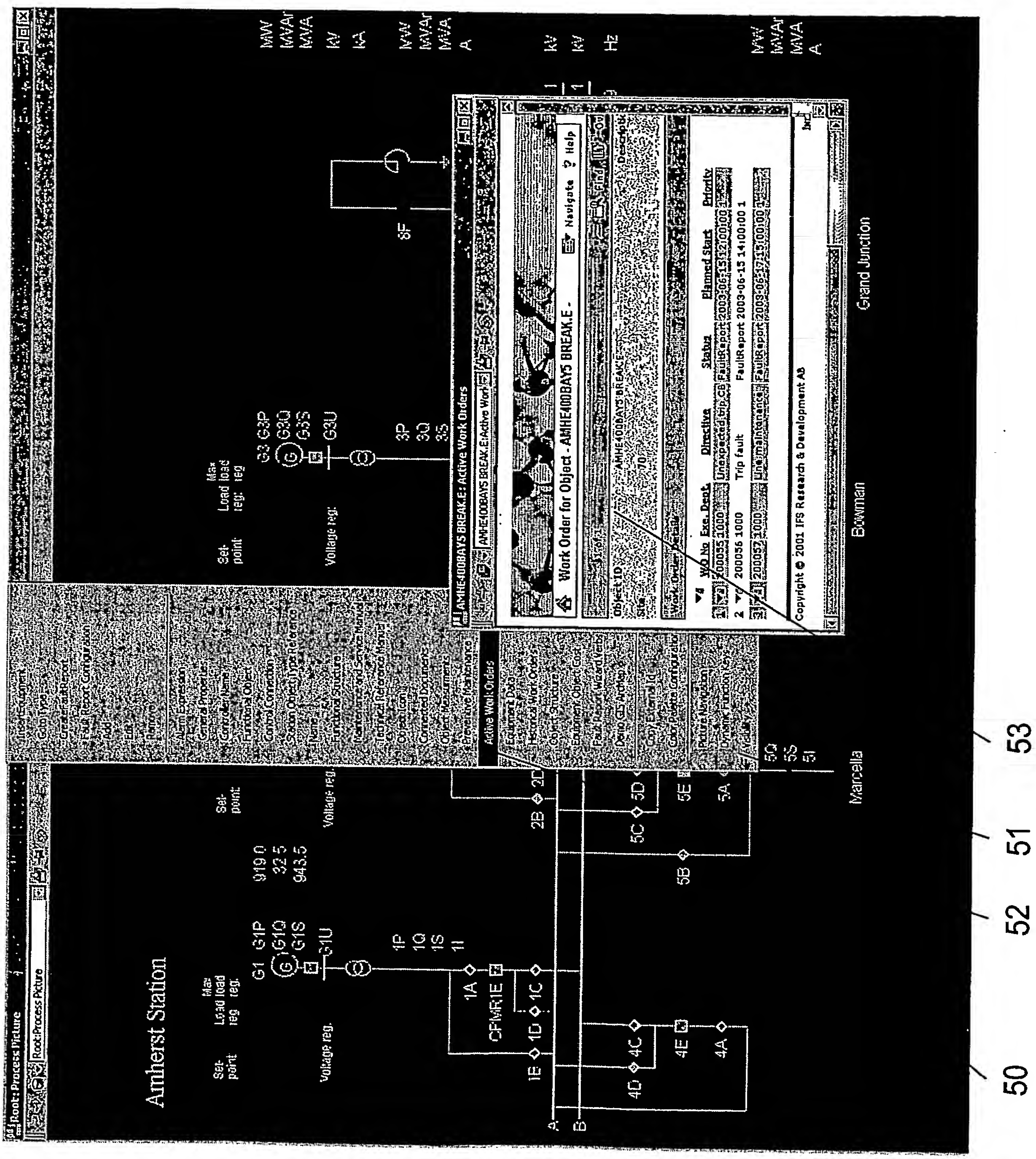


Figure 6

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

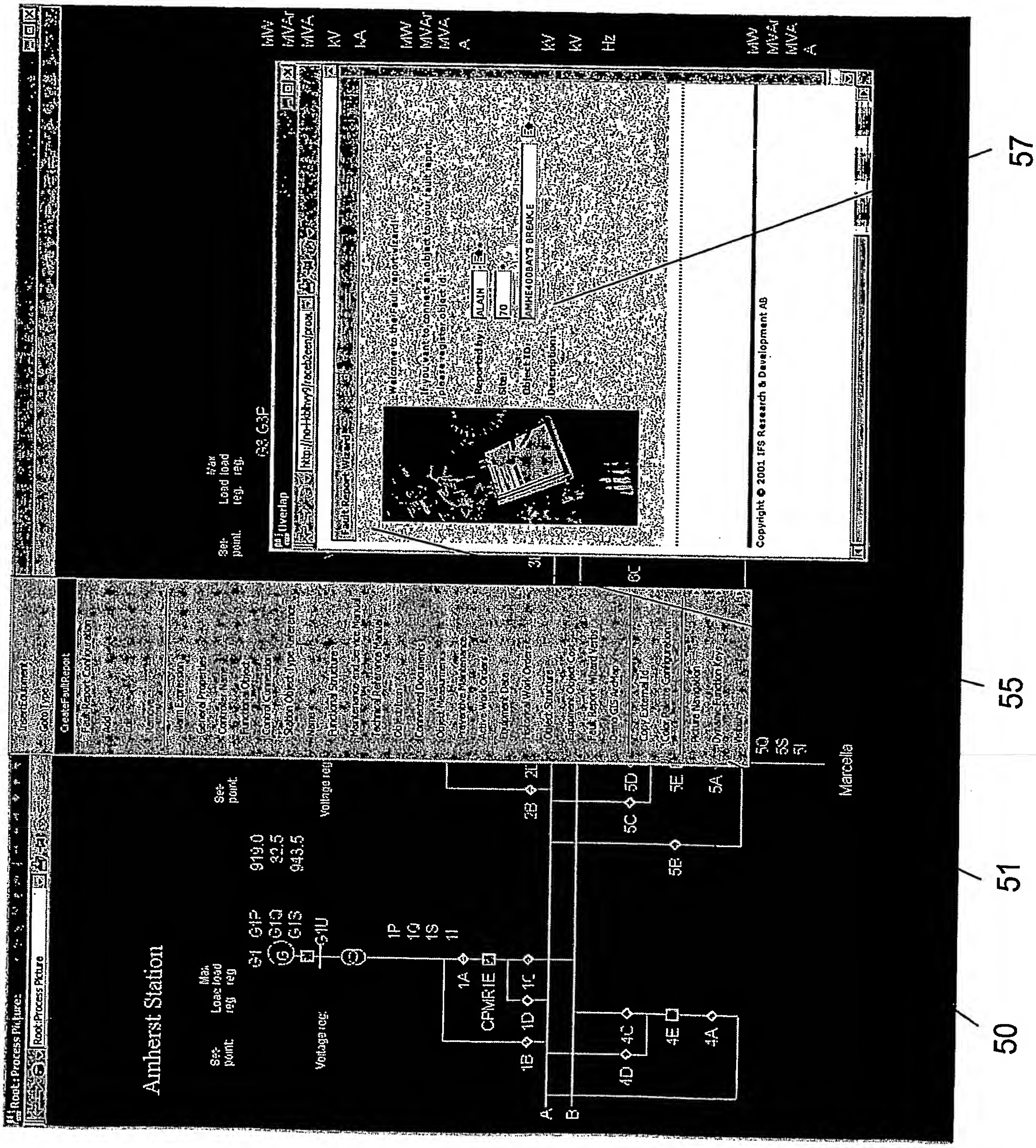
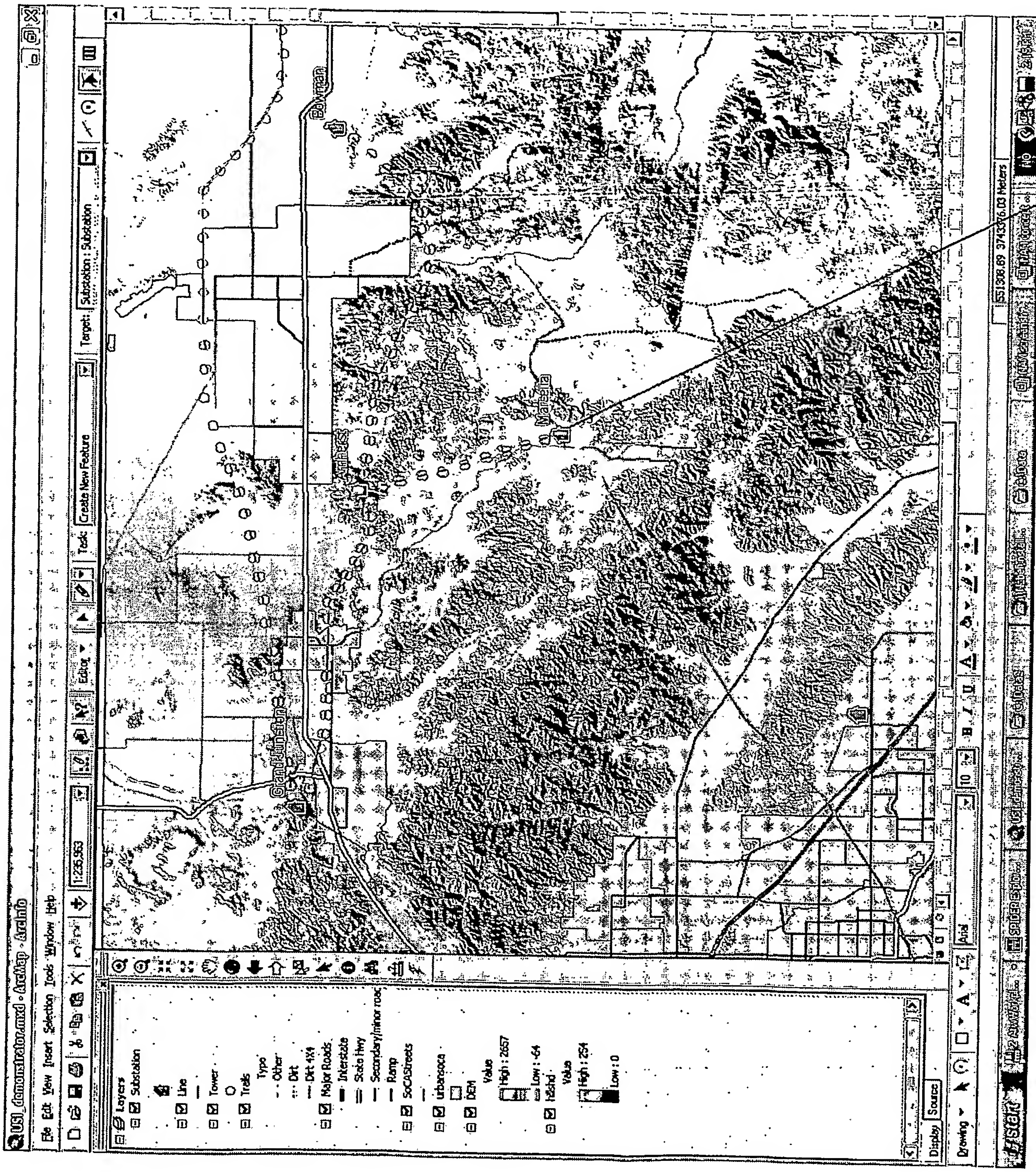


Figure 7

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE



40'

Figure 8

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

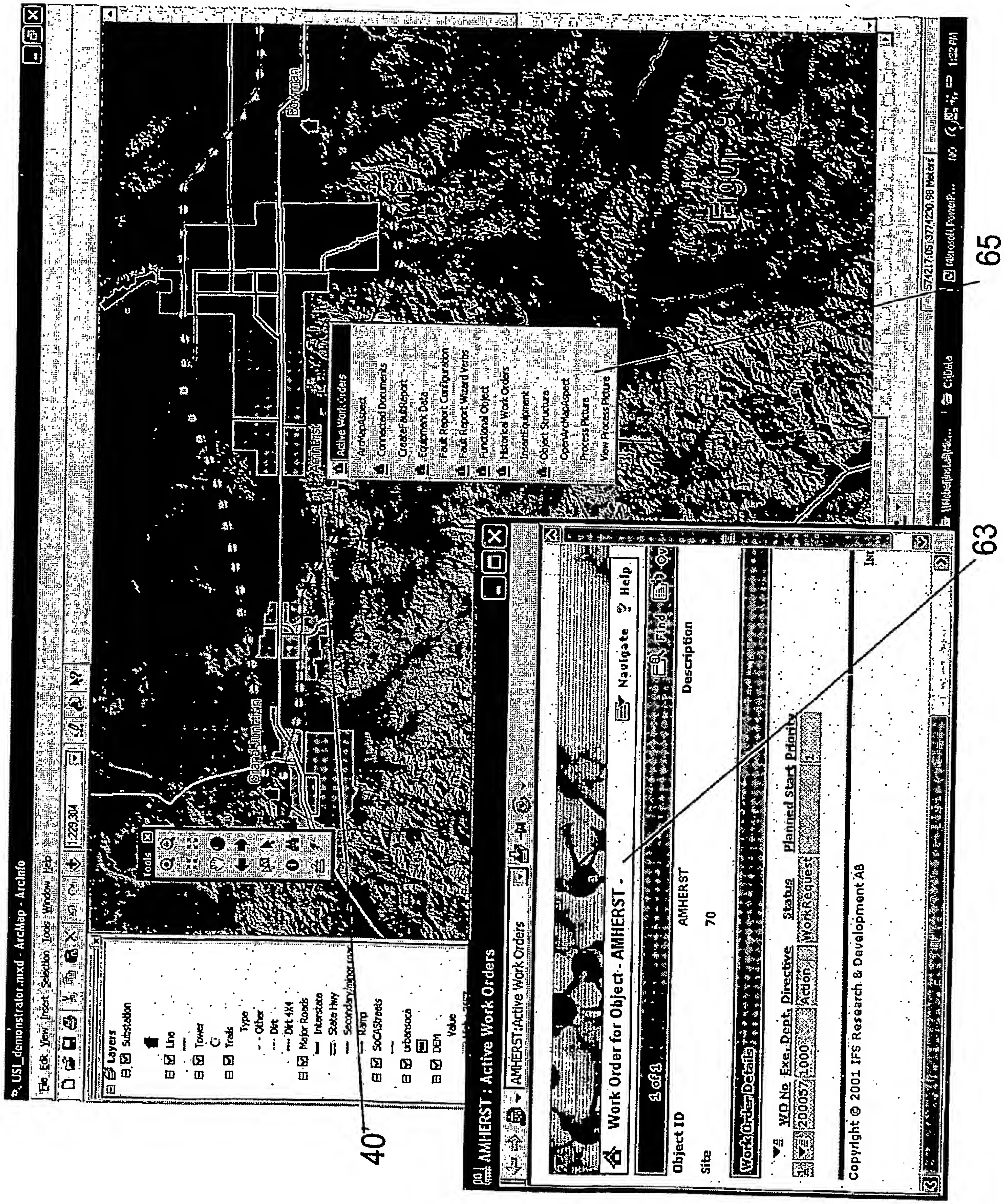


Figure 9

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

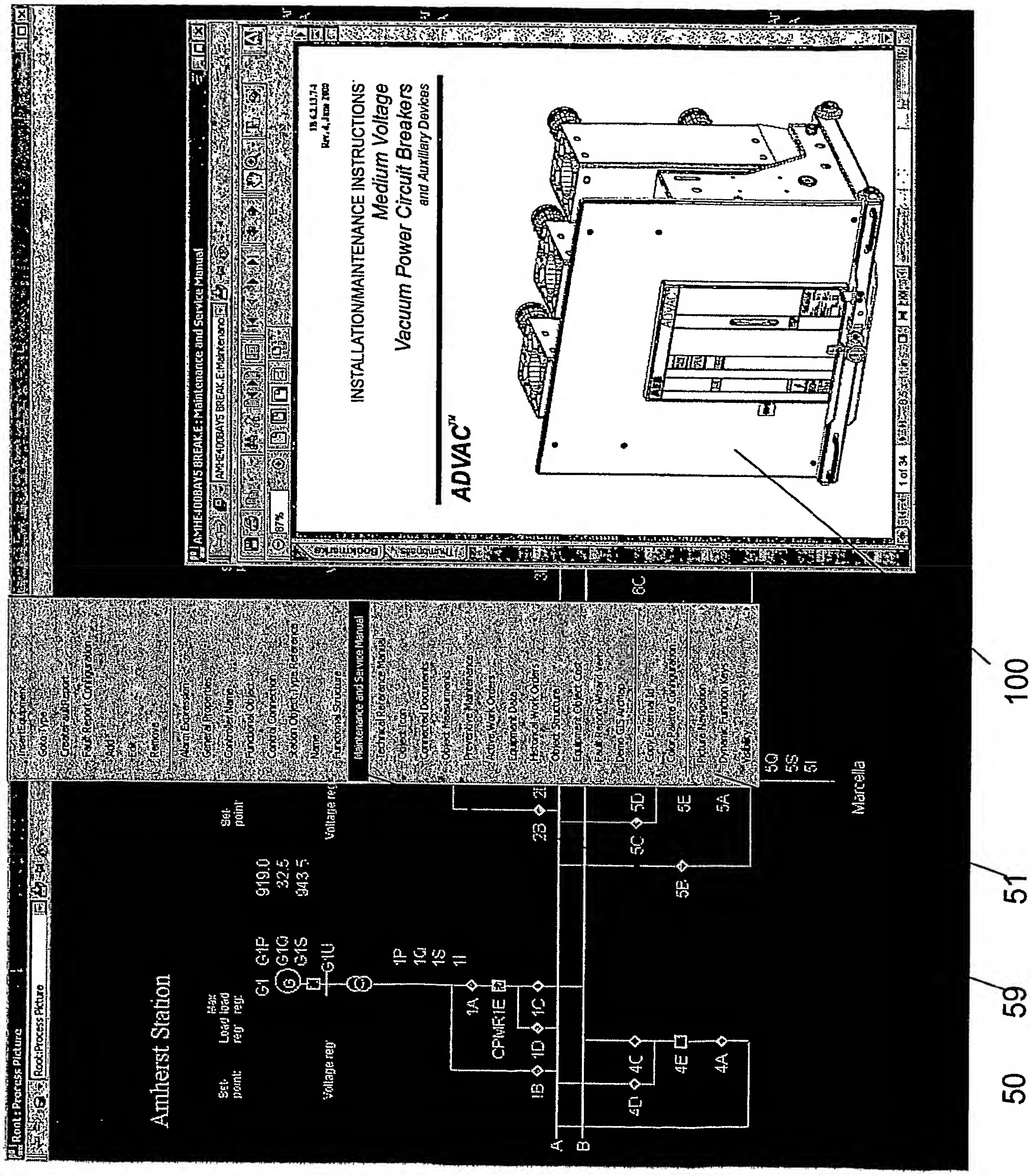


Figure 10

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

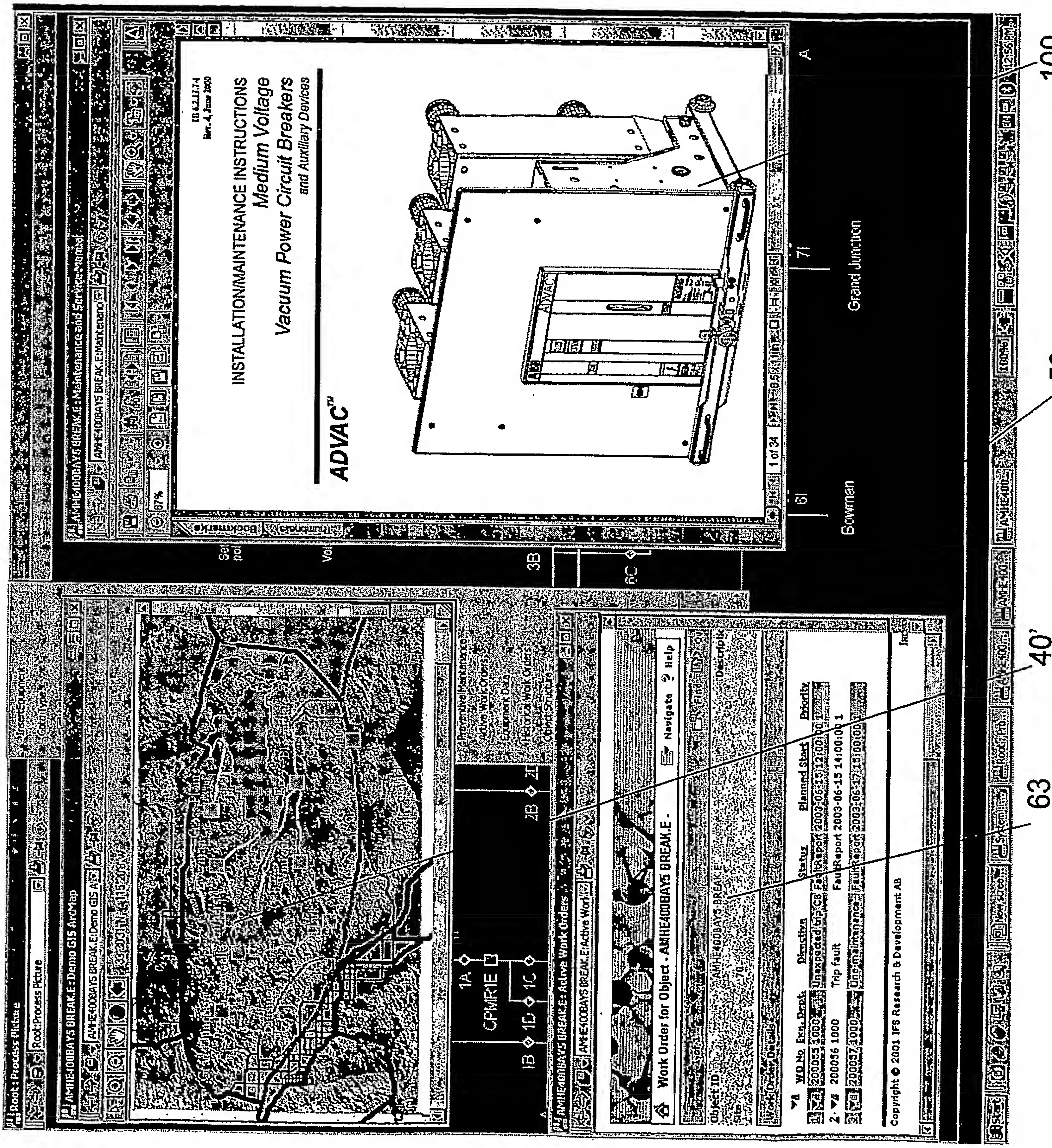


Figure 11

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

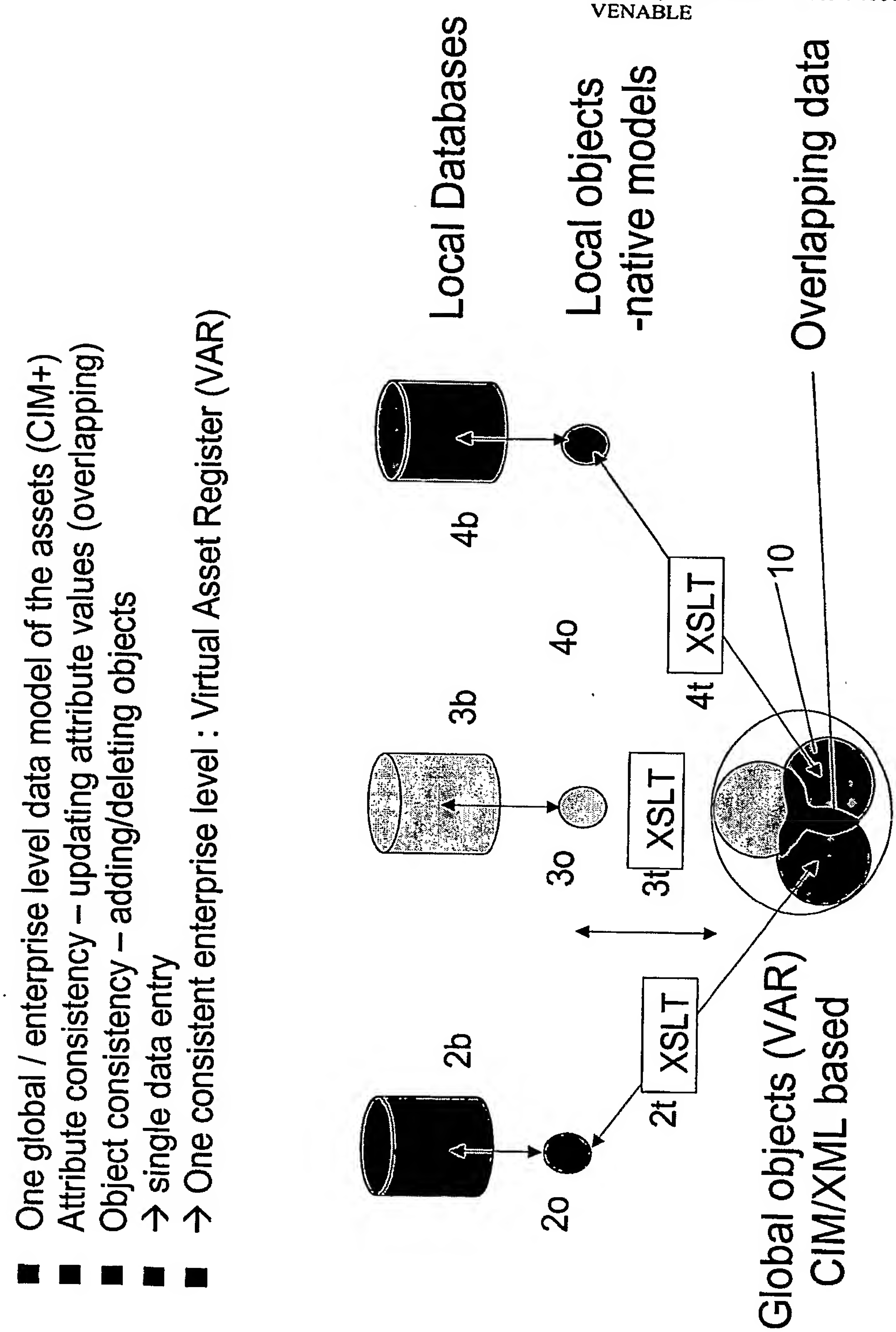


Figure 12

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

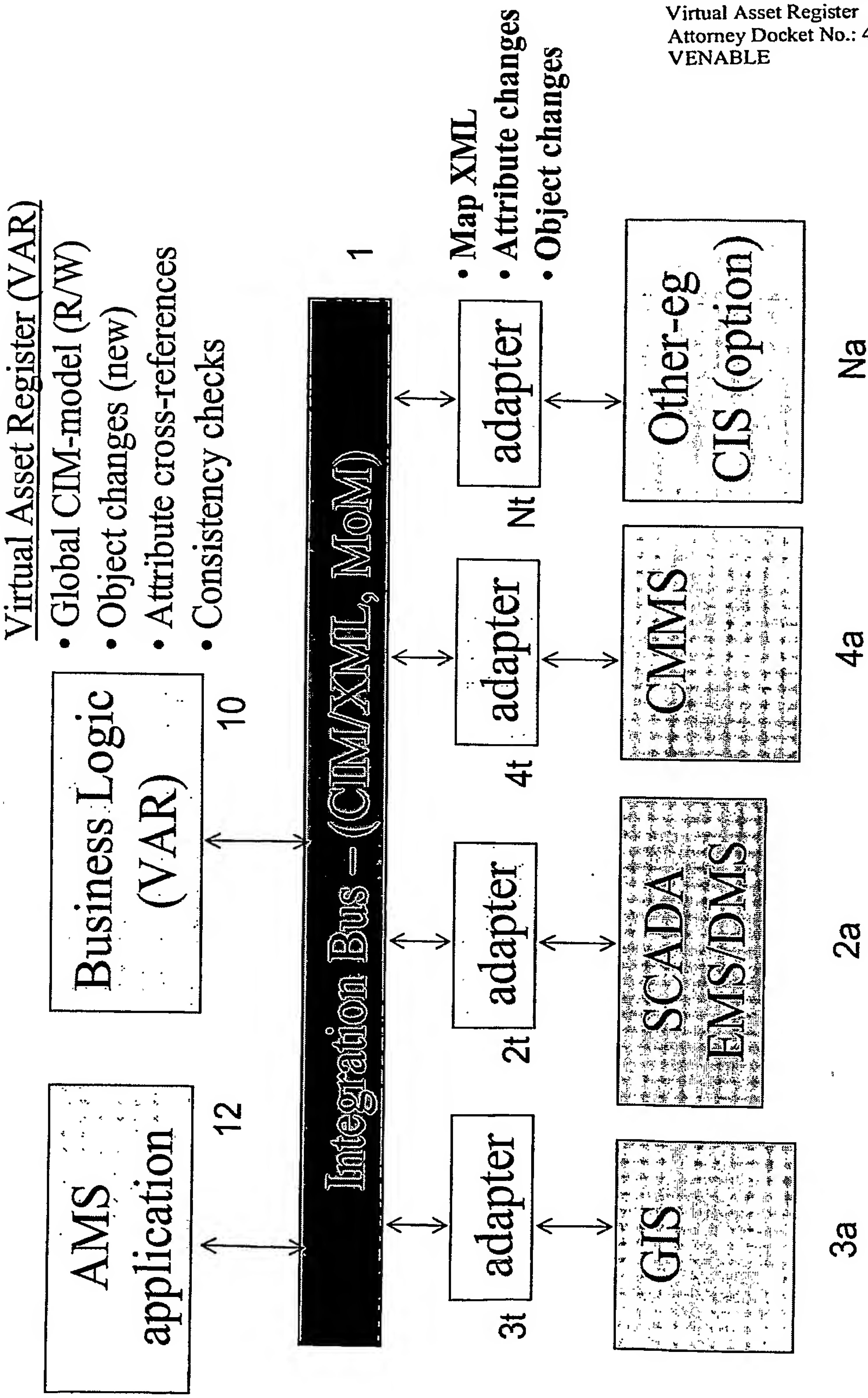


Figure 13

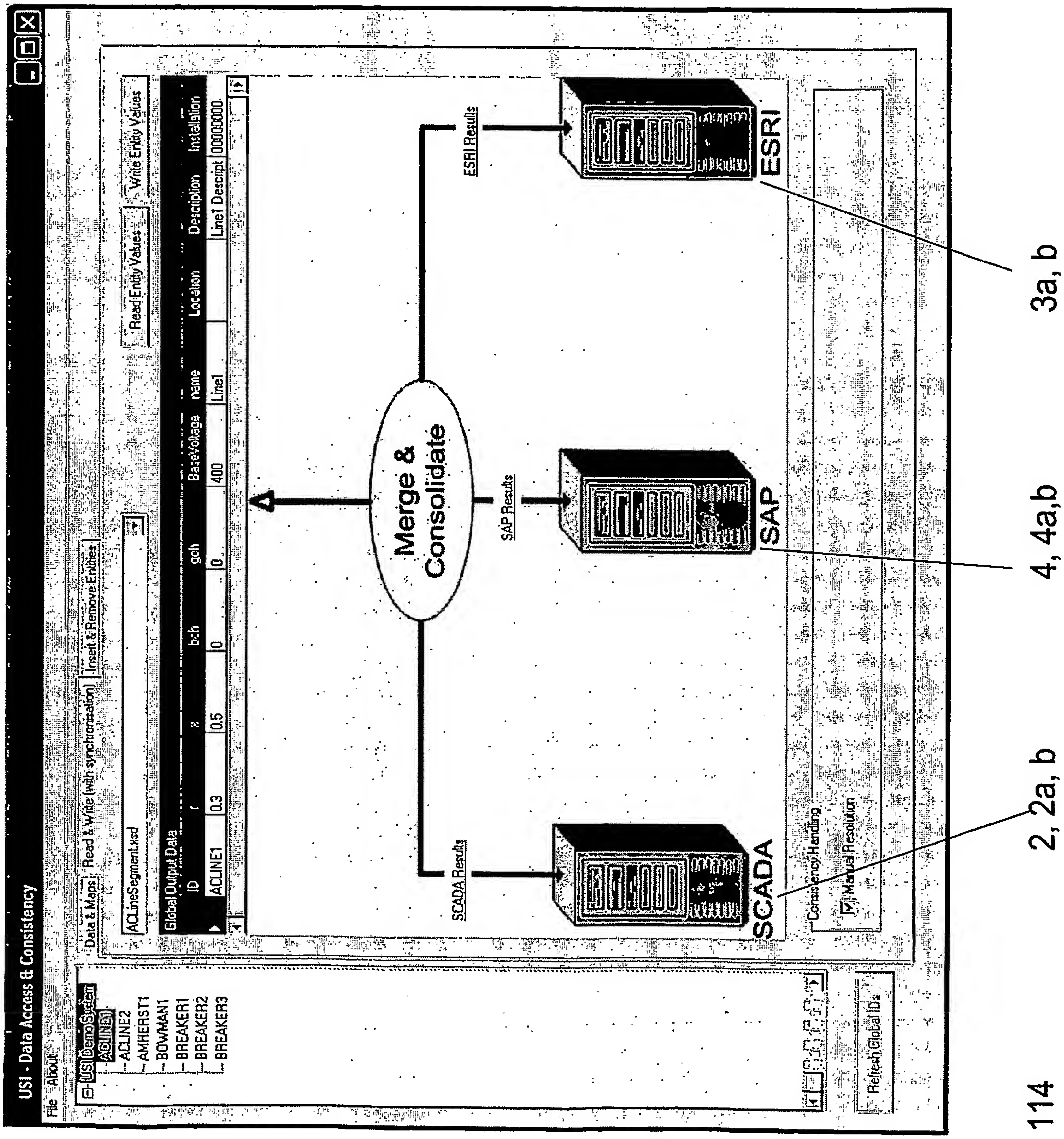

USI - Data Access & Consistency
File
About
USI Demo System
ACLINE1
ACLINE2
AMHERST1
BOWMAN1
BREAKER1
BREAKER2
BREAKER3
Refresh Global IDs
Data & Maps
Read & Write (with synchronisation)
Insert & Remove Entities
ACLineSegment.xsd
Read Entity Values
Write Entity Values
Global Output Data
ID
r
x
bch
gch
BaseVoltage
name
Location
Description
Installation
ACLINE1
0.3
0.5
0
0
400
Line1
Line1 Descript
00000000
Merge & Consolidate
SCADA Results
SAP Results
ESRI Results
SCADA
SAP
ESRI
Consistency Handling
Manual Resolution
2, 2a, b
4, 4a,b
3a, b
114

Figure 14

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

Mapping between local and global (CIM) data models

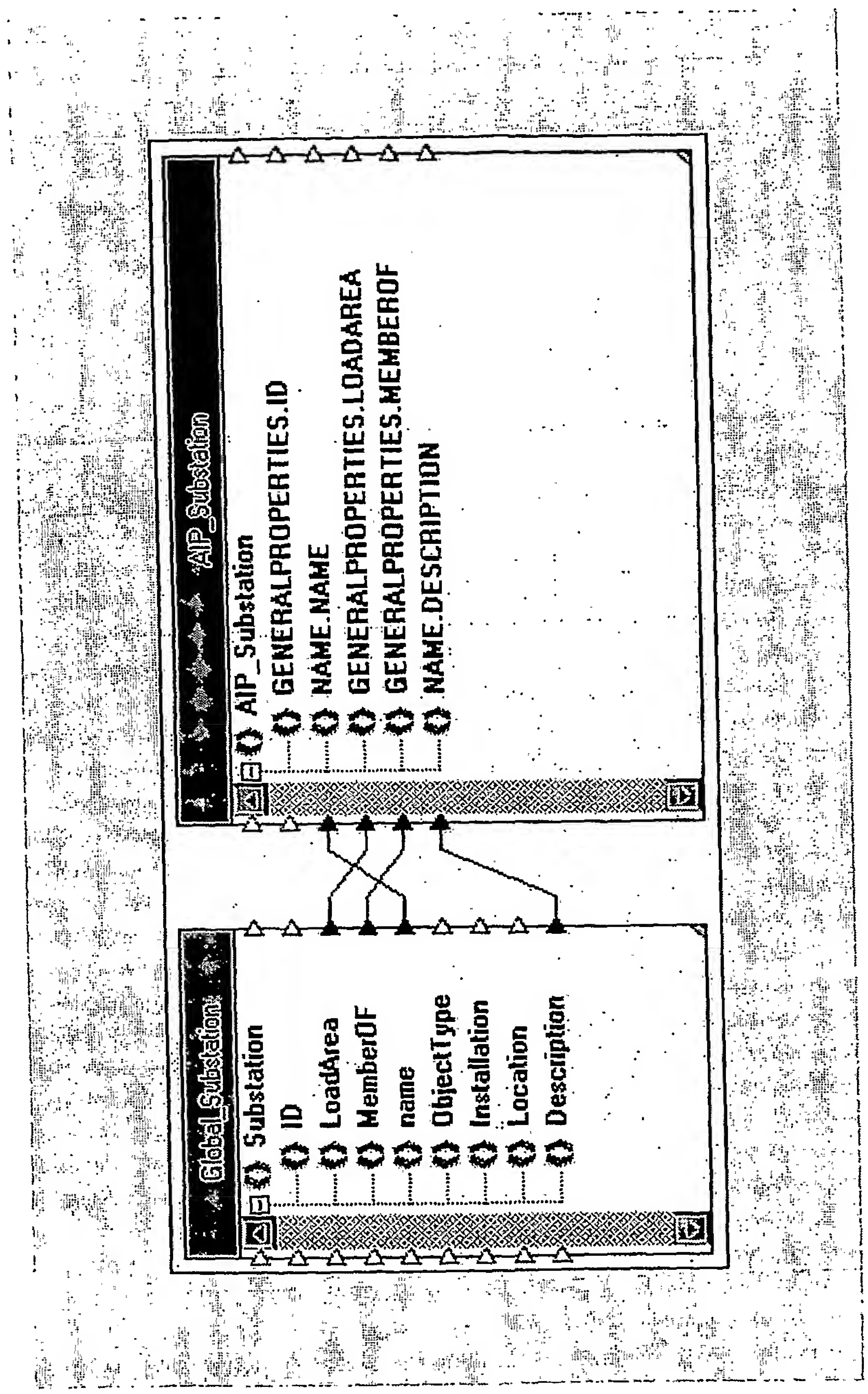


Figure 15

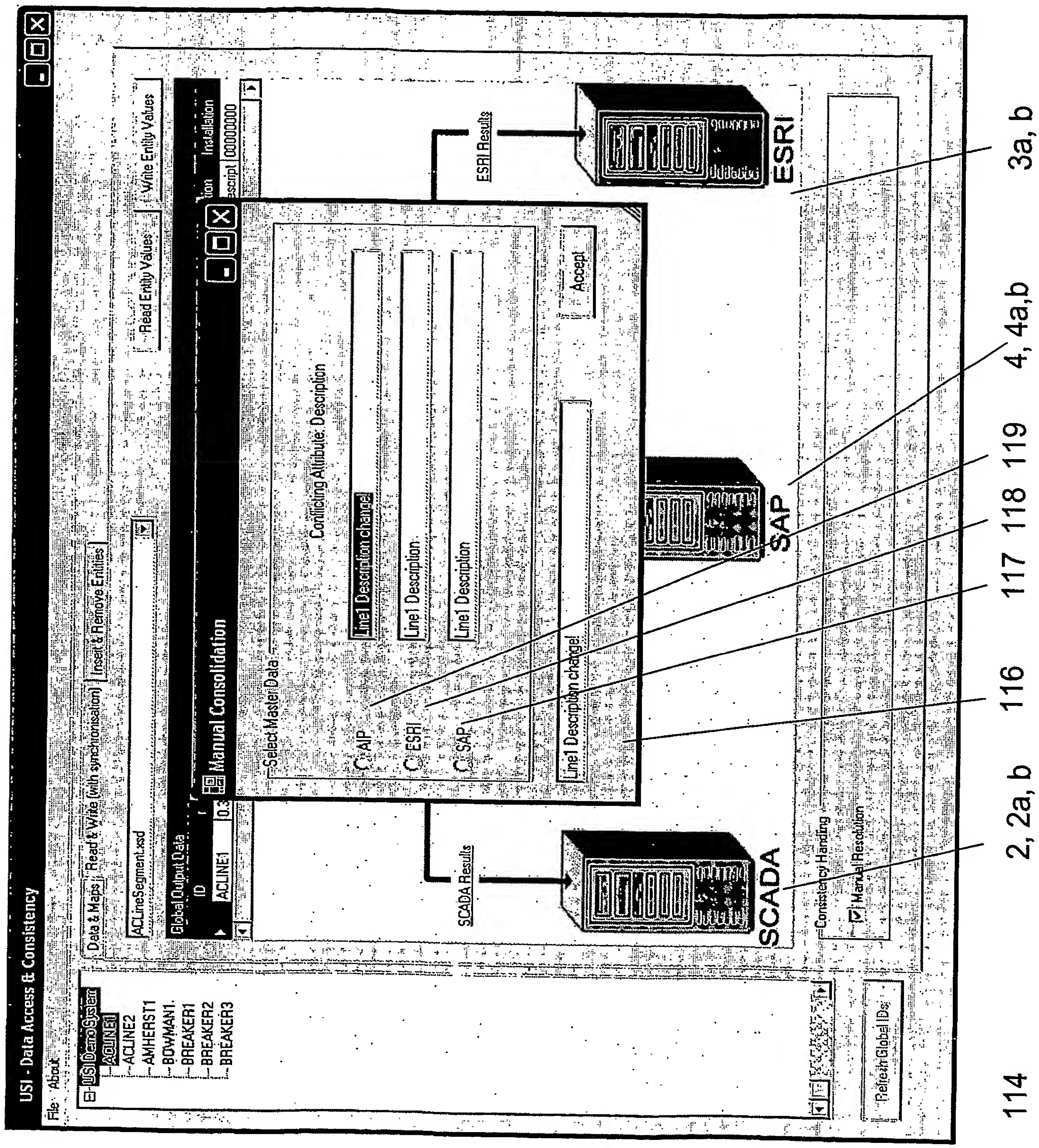

USI - Data Access & Consistency
File About
USI Demo System
ACLINE1
ACLINE2
AMHERST1
BOWMAN1
BREAKER1
BREAKER2
BREAKER3
Refresh Global IDs
Data & Maps
Read & Write (with synchronisation)
Insert & Remove Entities
ACLineSegment.xsd
Read Entity Values
Write Entity Values
Global Output Data
ID
ACLINE1
Installation
Manual Consolidation
Select Master Data
Conflicting Attribute: Description
AIP
ESRI
SAP
Line1 Description change!
Line1 Description
Line1 Description
Line1 Description change!
Accept
SCADA Results
ESRI Results
SCADA
SAP
ESRI
Consistency Handling
Manual Resolution
114
2, 2a, b
116
117
118
119
4, 4a,b
3a, b

Figure 16

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

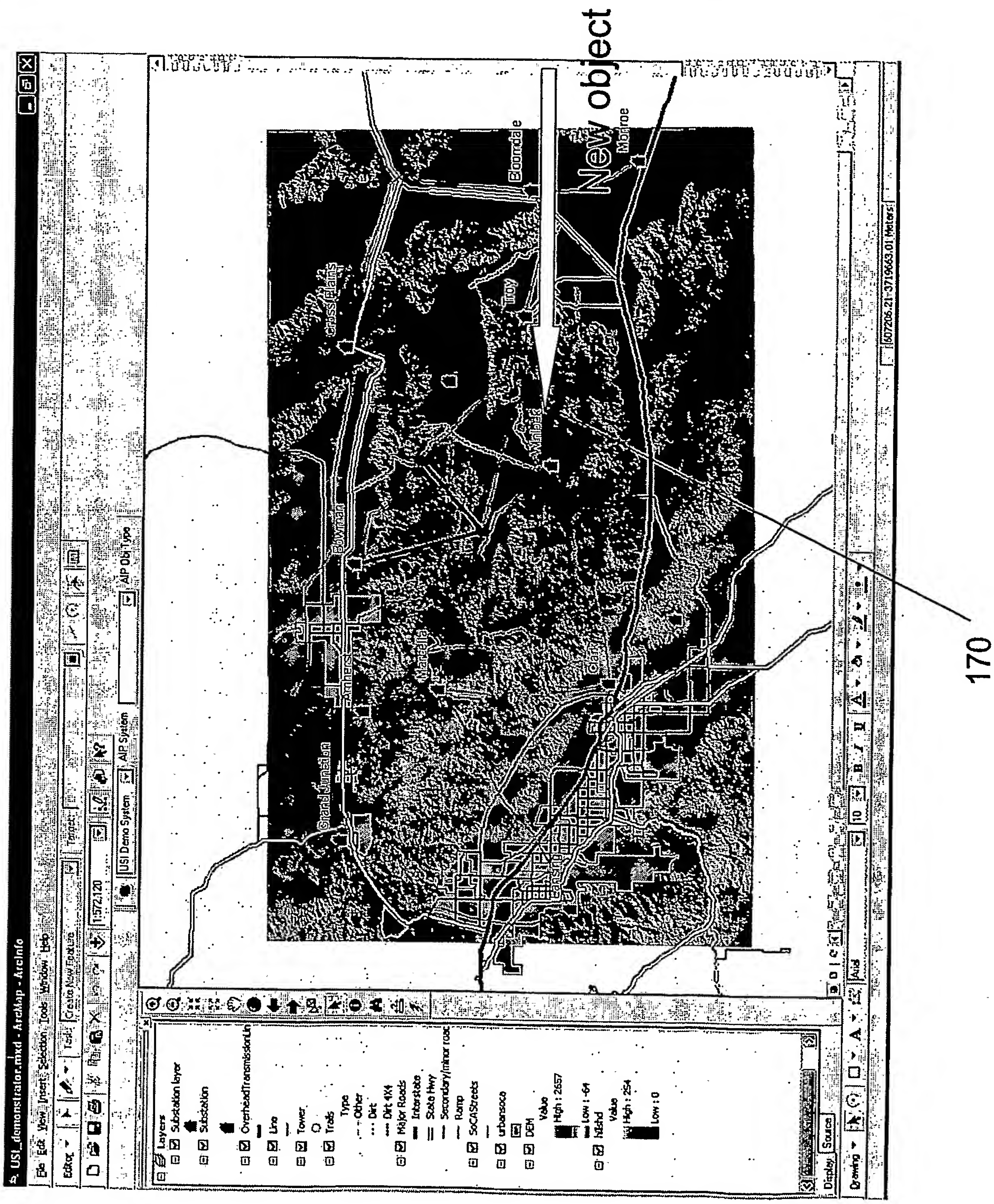


Figure 17

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

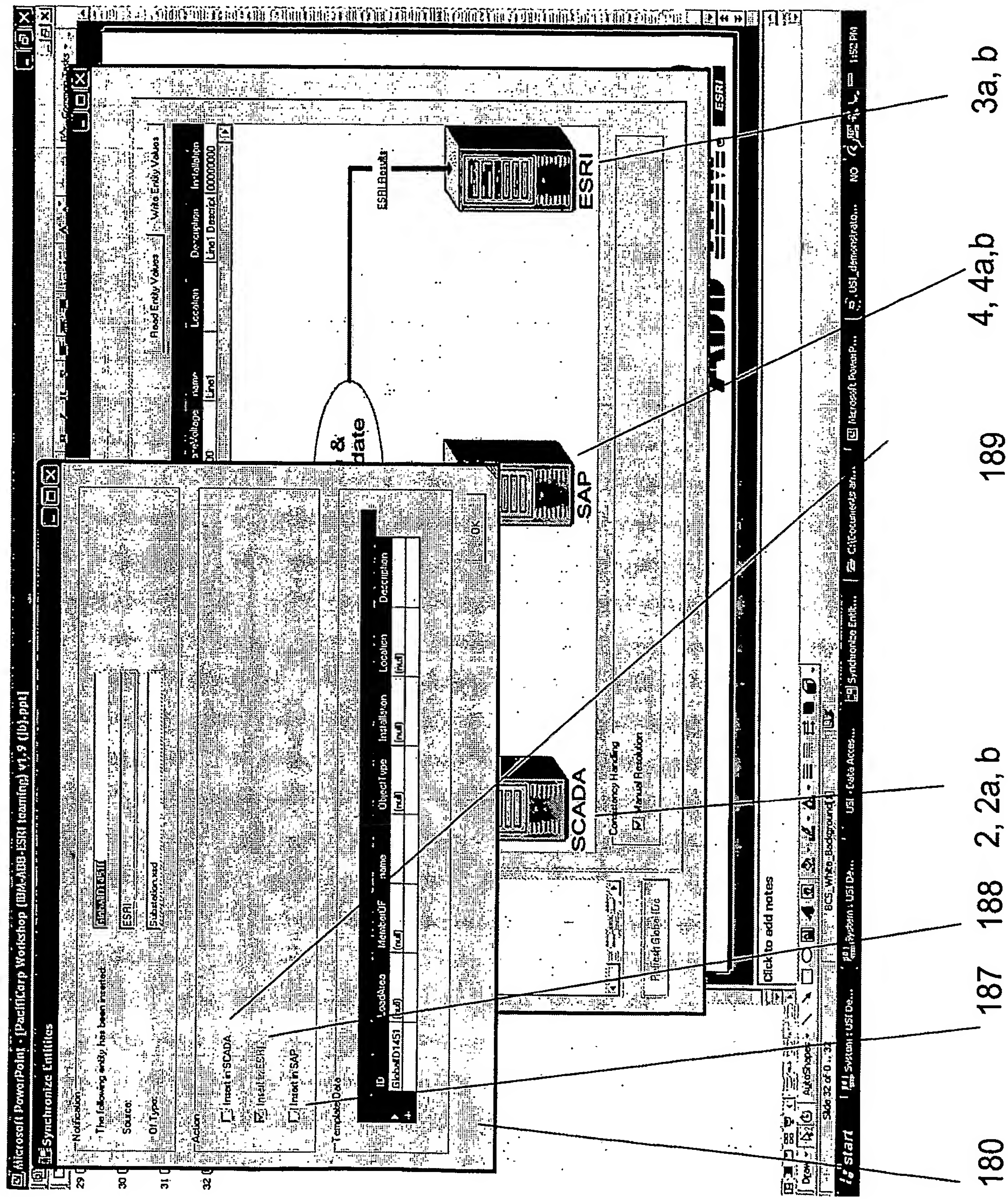


Figure 18

Inventor: Gorme SANDE et al.
Title of Invention: Method, System and Virtual Asset Register
Attorney Docket No.: 43315-9468US
VENABLE

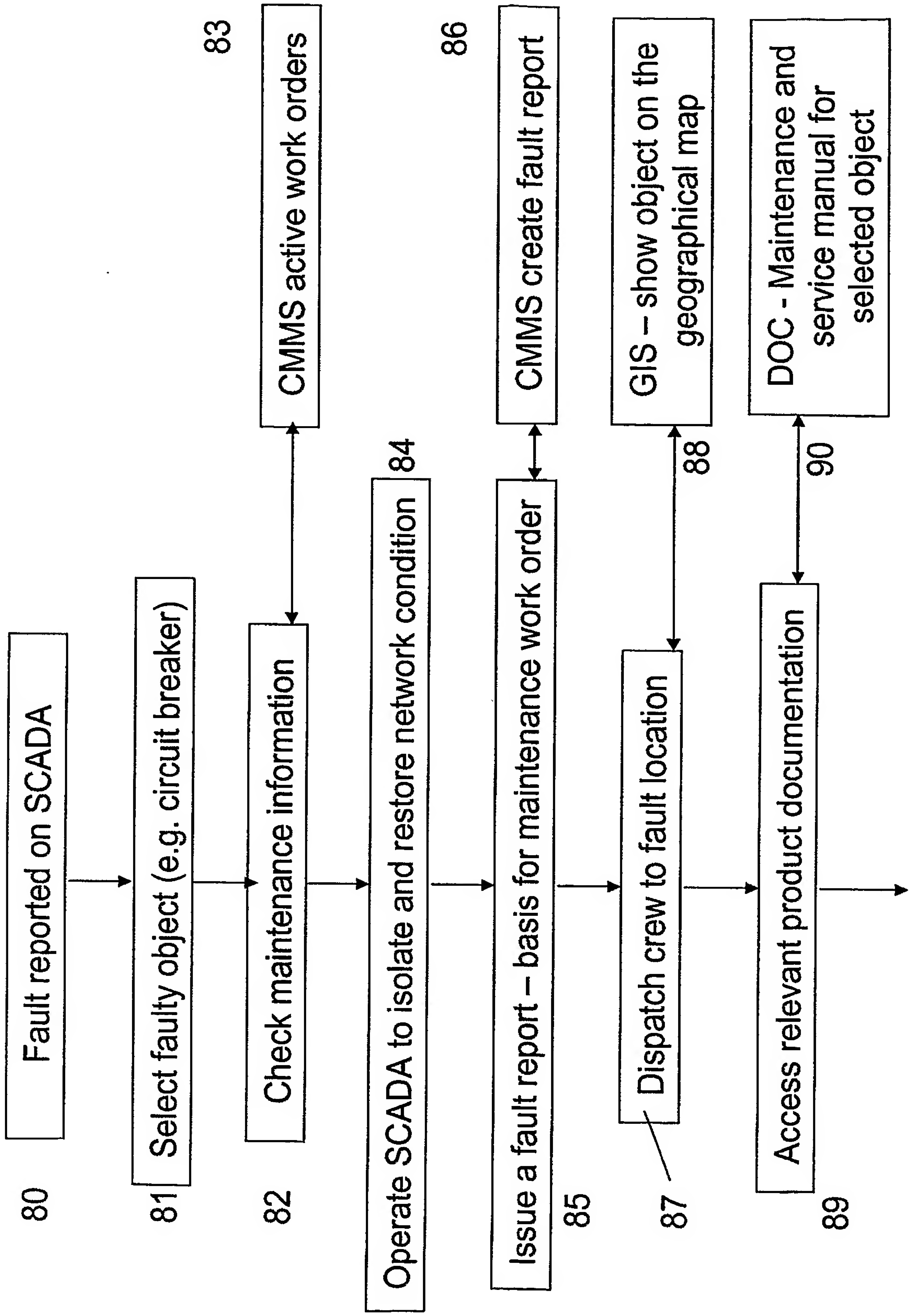
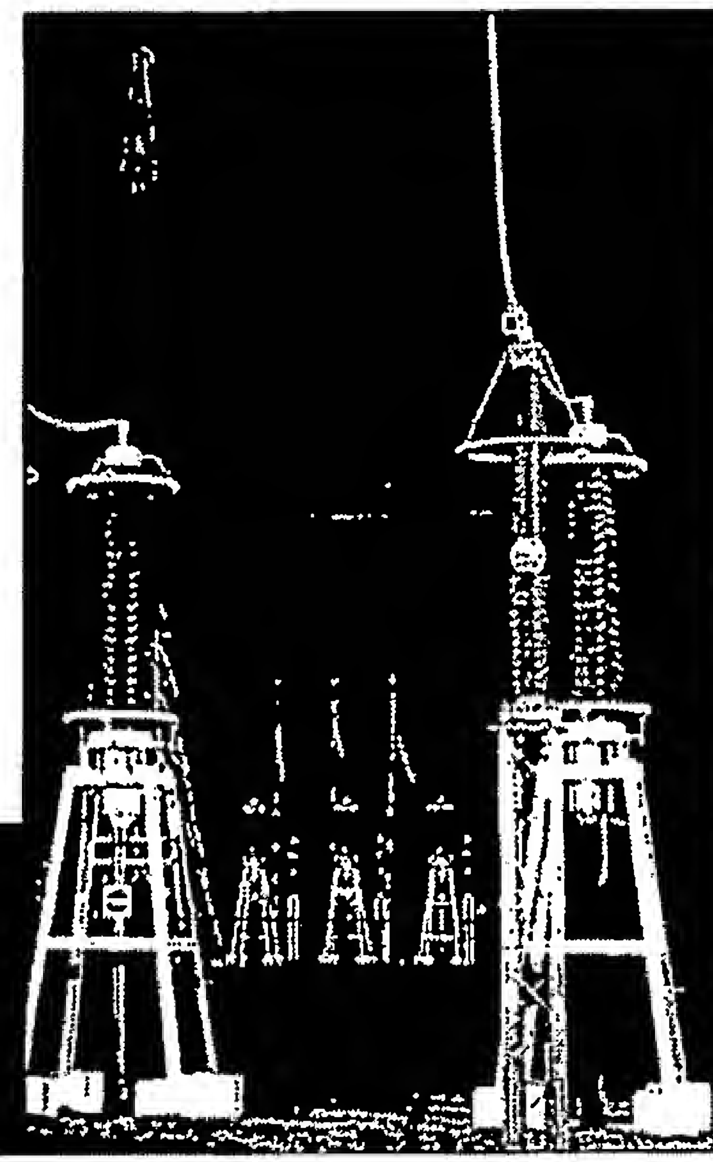


Figure 19

Jan Bugge

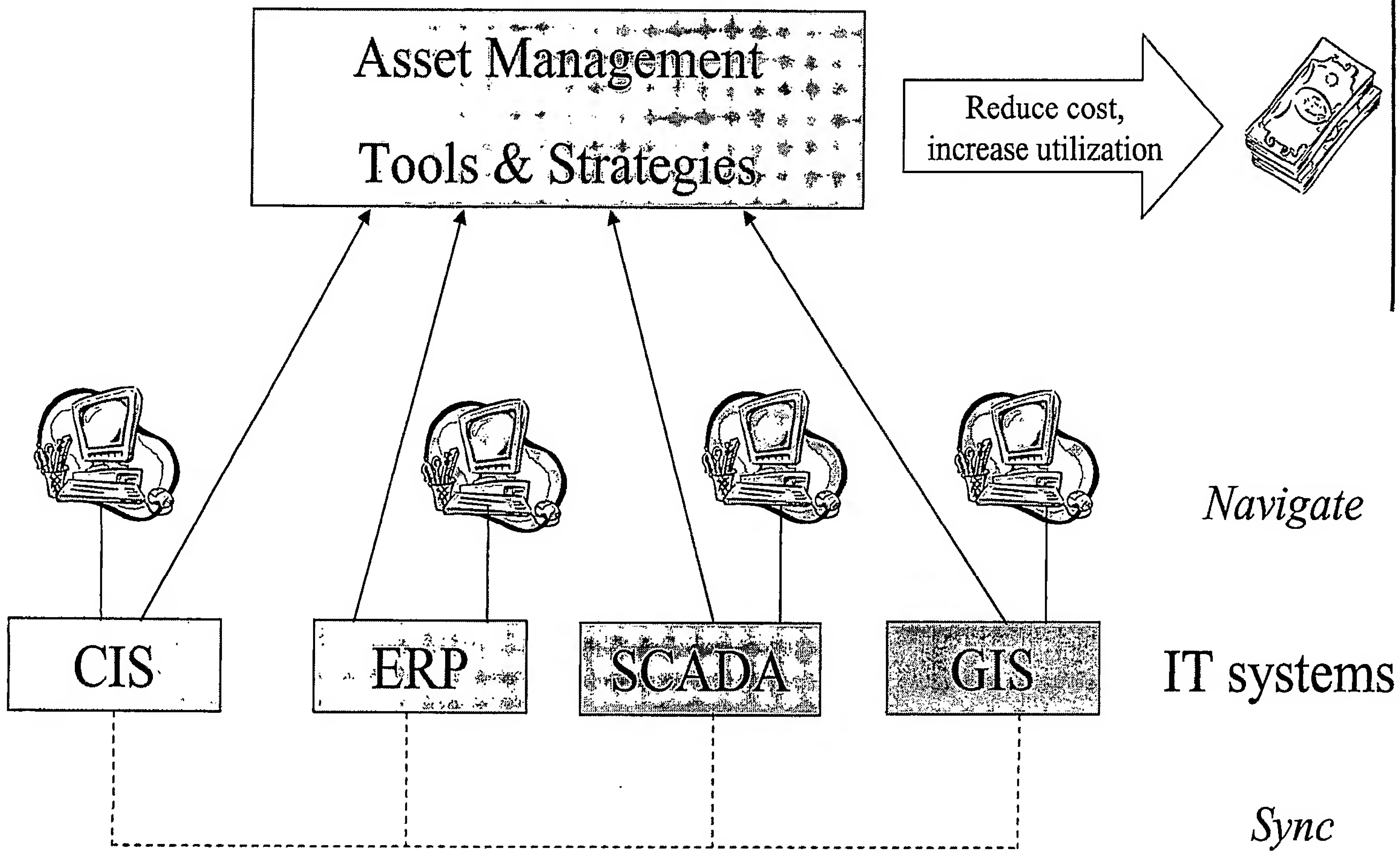


Integrated Network Asset Management

NAM Workshop

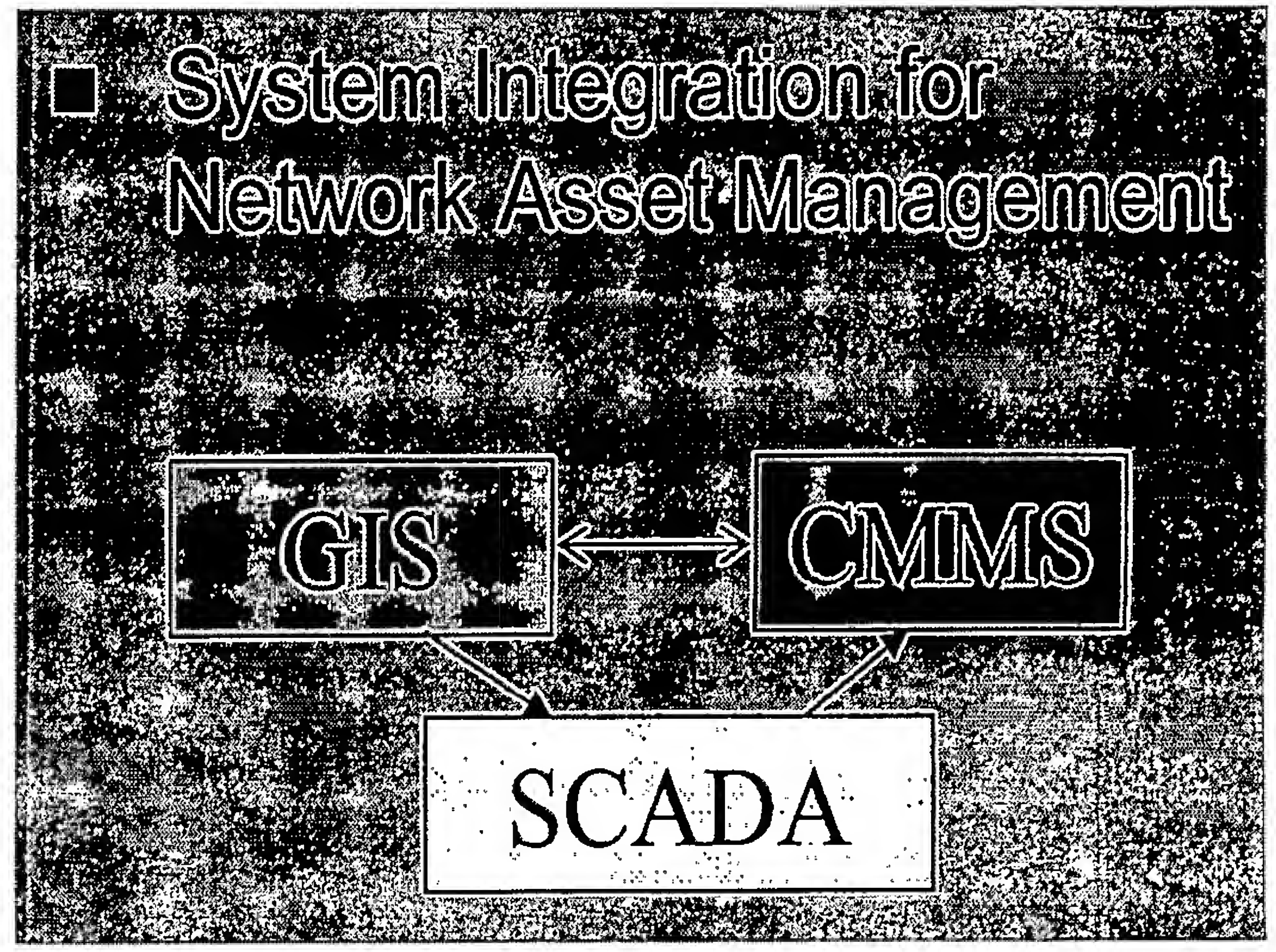
ABB

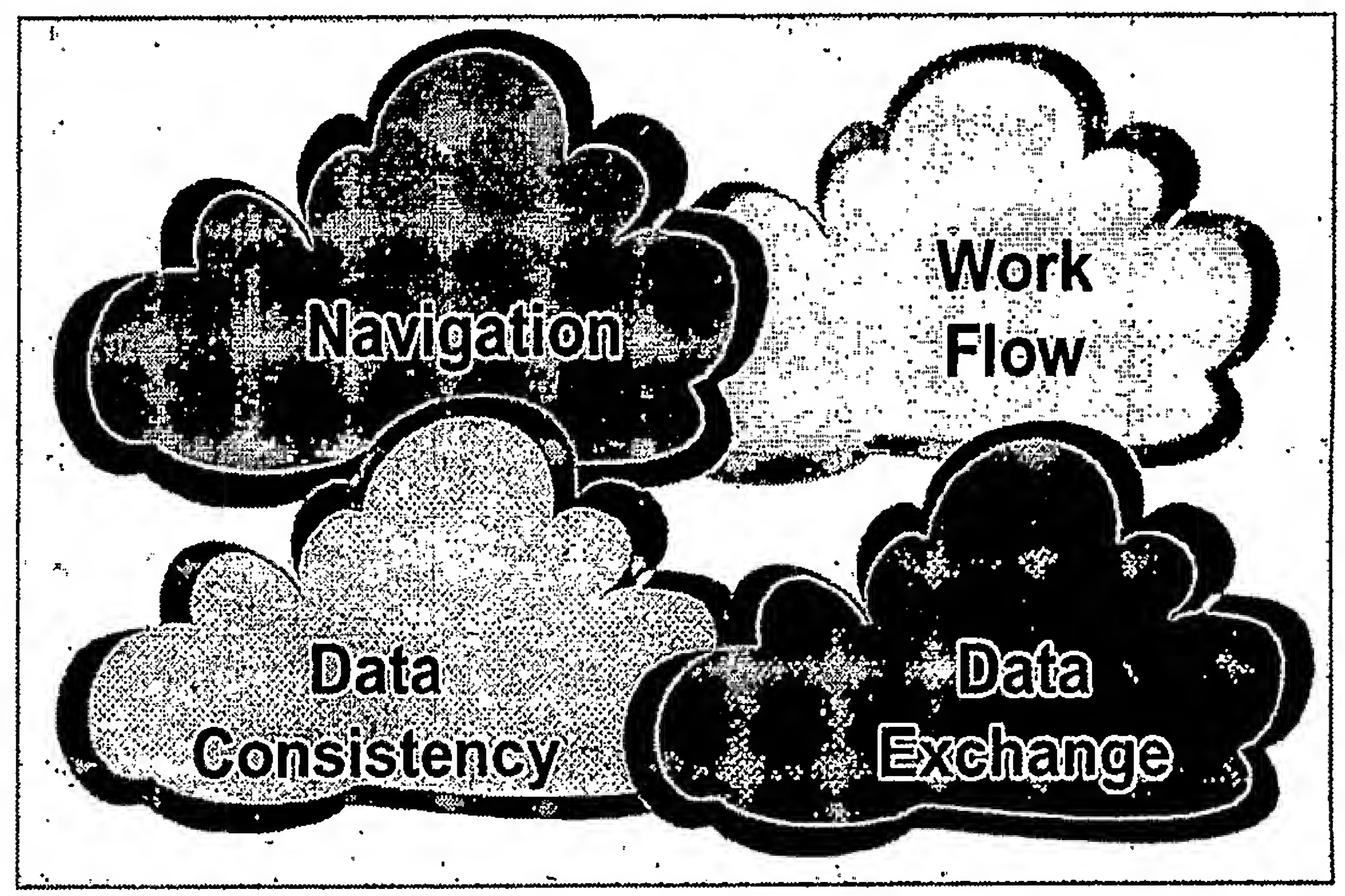
Network Asset Management – the big picture



Market trends – customer values

- Cost pressure
  - HR cost ~70% (~½ mobile)
  - Maintenance planning
  - Data maintenance
- M&A – larger utilities
  - Less control centers
  - IT consolidation
- Power system reliability
  - Correct information
  - Fast decisions
- Asset optimization
  - Life cycle assessment
  - Condition monitoring
  - Dynamic loading





Integrated Network Asset Management

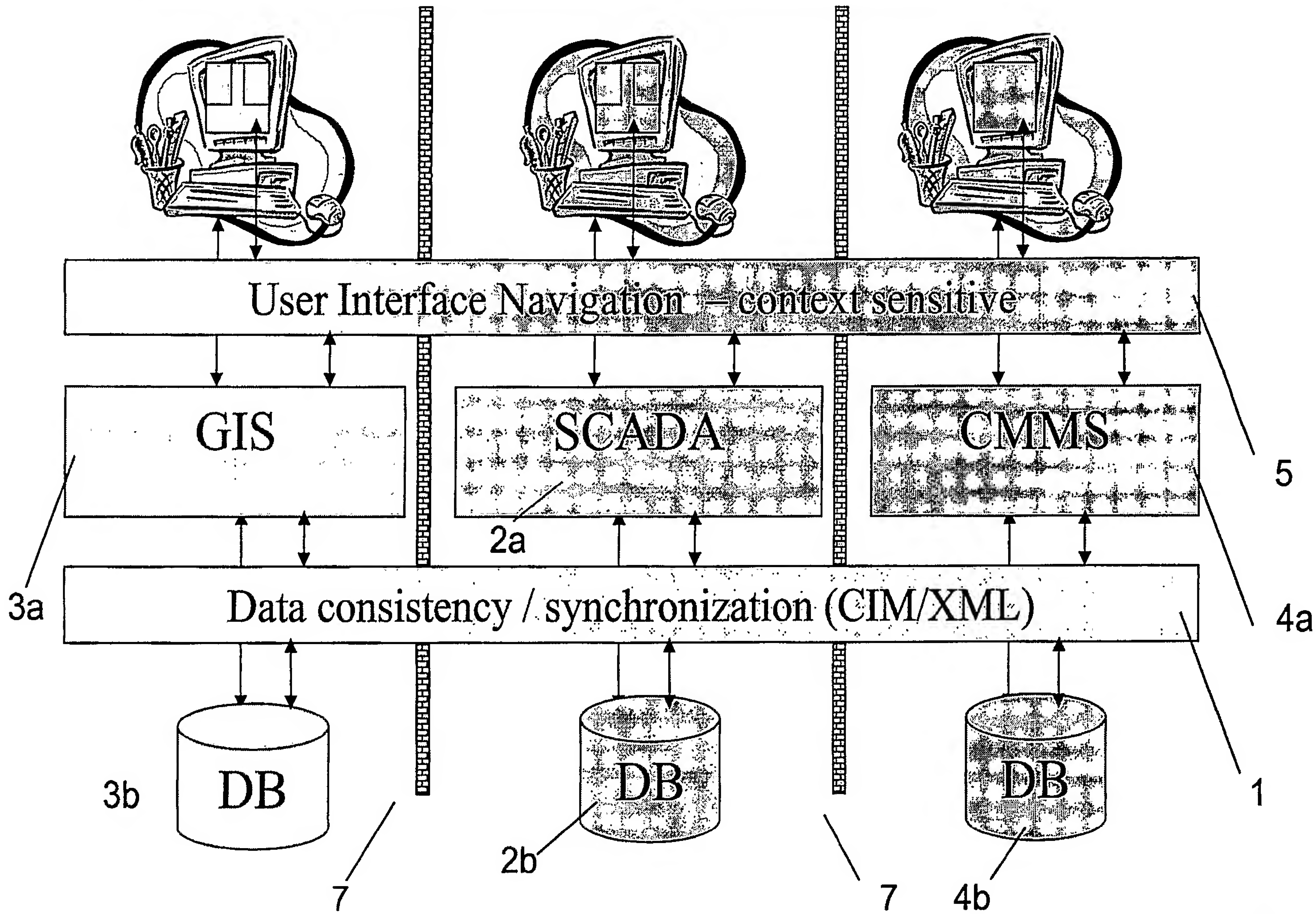
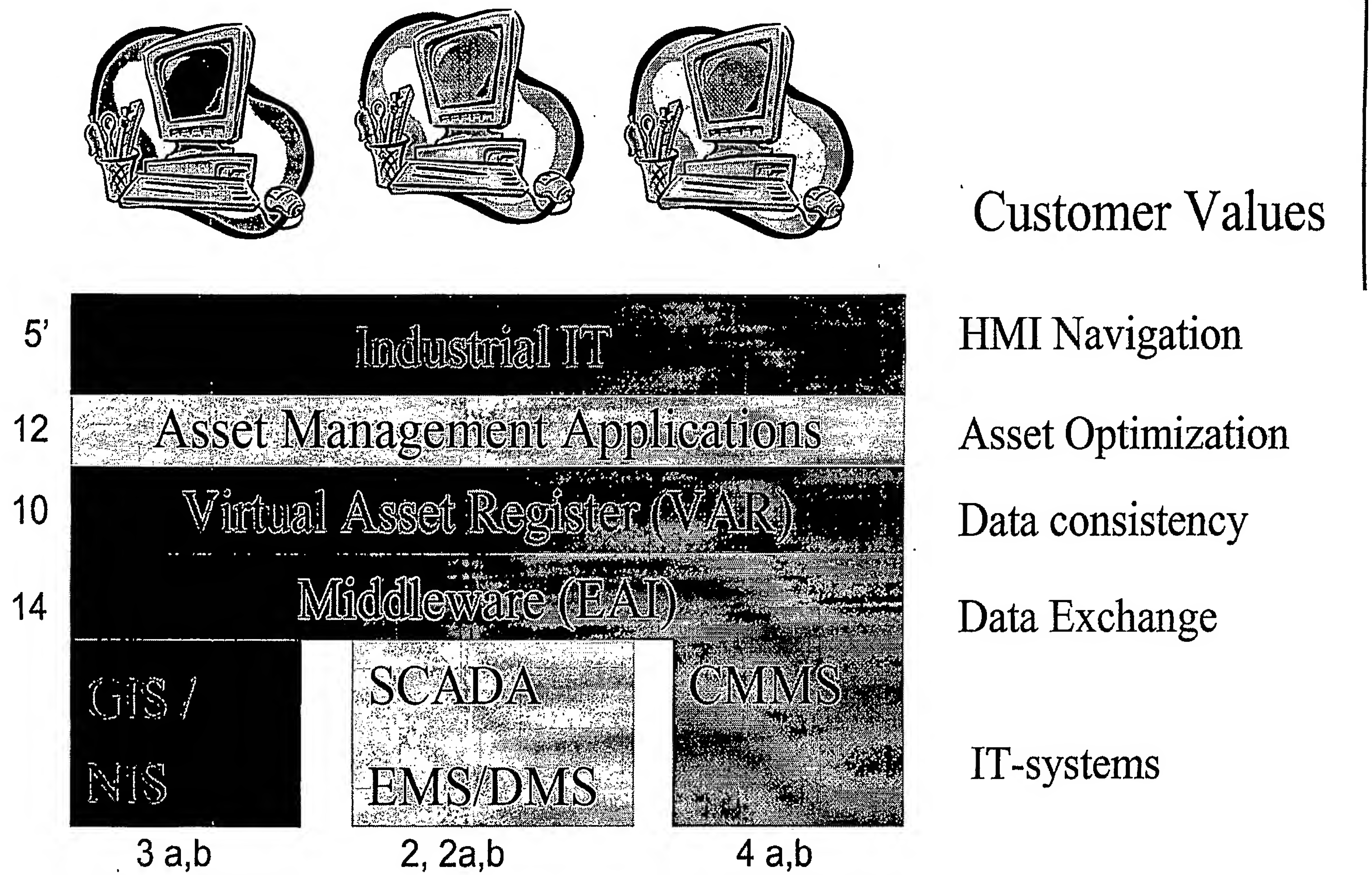


Figure 1

Advantages by the integrated solution

- User Interface Navigation (IIT)
  - Timely access to all relevant information
    - From all integrated IT systems
  - Context sensitive
    - Navigate between aspects or views on same object
- One consistent asset representation (VAR)
  - Single data entry
  - Automatic synchronization
    - data exchange between applications
  - Consistency checks
  - Mapping to CIM/XML model
    - Import & export to CIM/XML
    - Applications can work on generic CIM model
- Reduced data maintenance cost
- Increased quality of asset data
- Improved decision support

System Integration concept



- Solutions that use standard products

Figure 2

Customer values in development steps

Development stage	Customer value	Comment	Mapping
Step 3	Asset optimization	Lifecycle, utilization	CIM/XML
Step 2	Consistency	Add/delete objects consistently	Structure
	Data exchange	Consistent values, CIM model	Attribute
Step 1	Navigation	Context sensitive, IIT (Aspect Object)	Object

Figure 3

Condition based Maintenance

- Operators have seamless access to the maintenance system from SCADA. They can create fault reports, access work order and spare part info, etc.
- Automation equipment can trigger work orders based on real time information from the equipment it self. Operational hours, fieldbus information, etc.

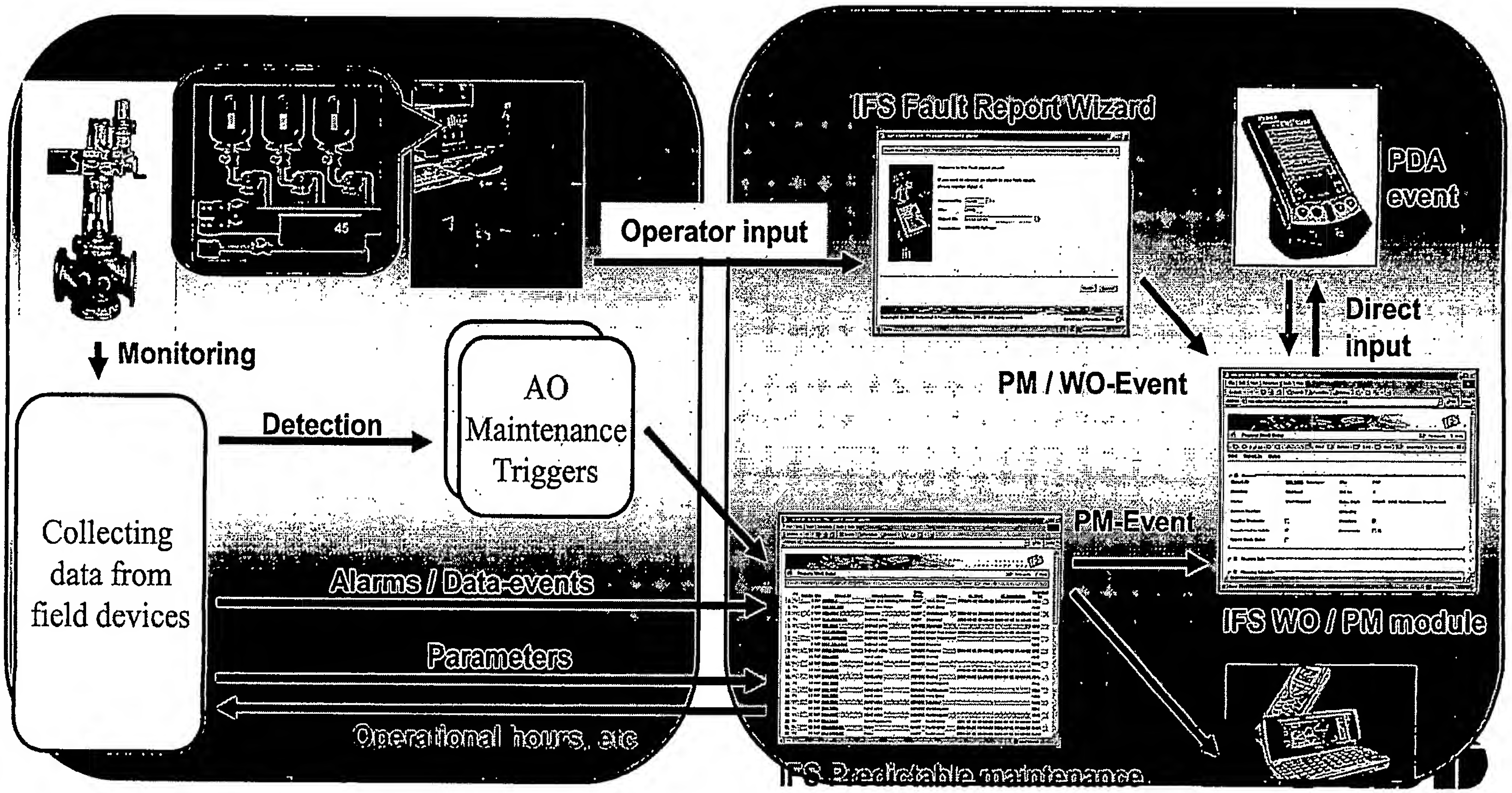


Figure 4

Market Drivers – customer values

- Reduced O&M cost
  - Reduced cost for data engineering
  - Condition based maintenance
  - Higher utilization of mobile crew

- Power System Reliability
  - Timely access to right information
    - All relevant data
    - Correct decisions

- Reduced Investment cost
  - Consistent asset data
    - improved asset utilization
  - Optimal replacement strategy

ABB

ABB

Cross Navigation Demo

System integration demo

- Key components
  - SCADA user interface (WS500)
  - IFS CMMS maintenance management
  - ESRI planning & mapping system (GIS)
  - Industrial IT (HMI integration)
- Functionality demonstrated
  - Seamless user interface integration – between SCADA, CMMS or GIS
    - e.g context sensitive access to CMMS from SCADA or GIS (object linked)

Demo – fault scenario

- Fault reported on SCADA
  - Select faulty object (e.g. circuit breaker) (PIC #1)
- Check maintenance information
  - CMMS active work orders (PIC #2)
- Operate SCADA to isolate and restore network condition
- Issue a fault report – basis for maintenance work order
  - CMMS create fault report (PIC #3)
- Dispatch crew to fault location
  - GIS – show object on the geographical map (PIC #4)
- Access relevant product documentation
  - DOC - Maintenance and service manual for selected object (PIC #5)

SCADA alarm, selecting object, list of aspects (#1)

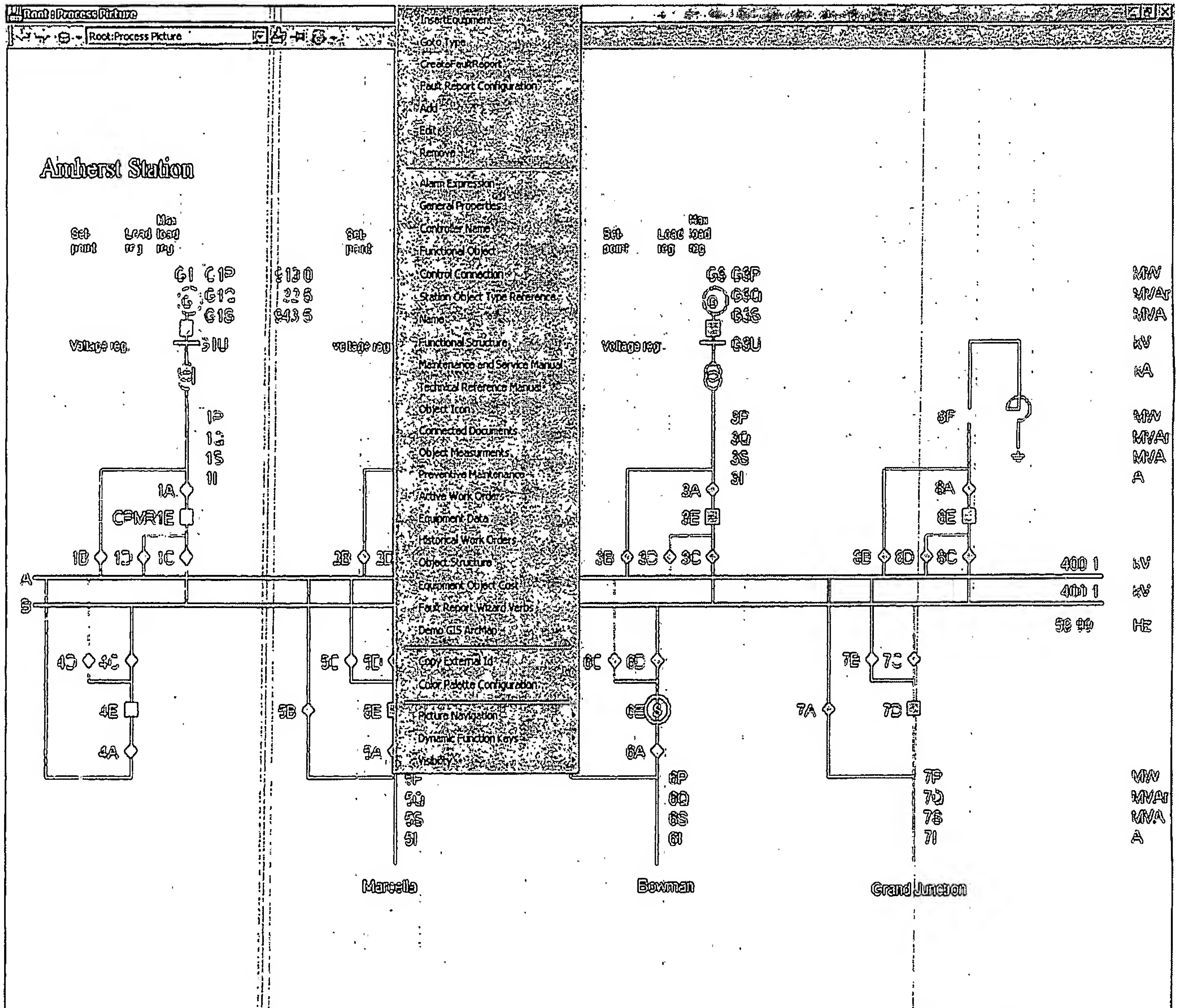


Figure 5

CMMS show active work orders (#2)

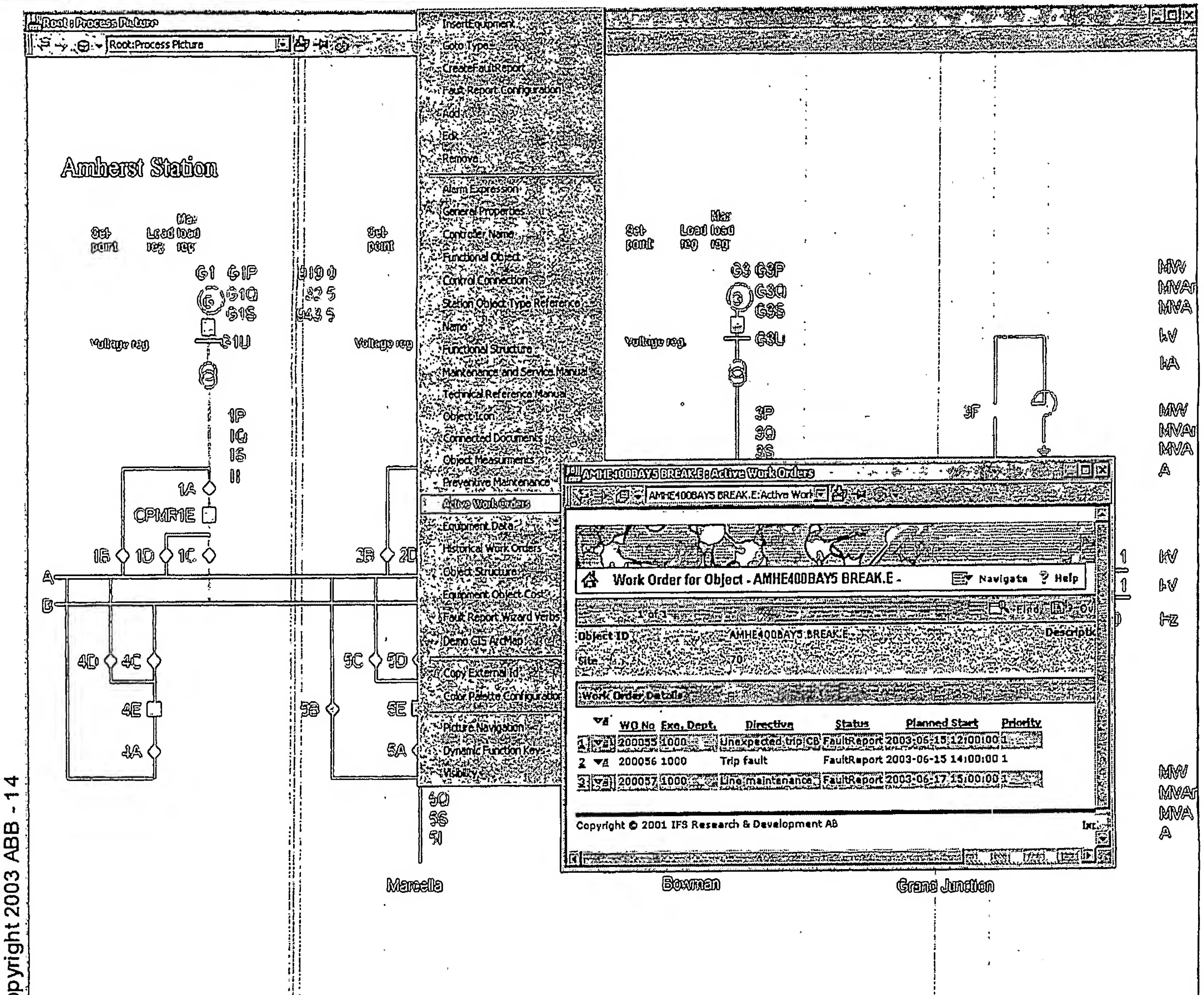


Figure 6

Create Fault Report (#3)

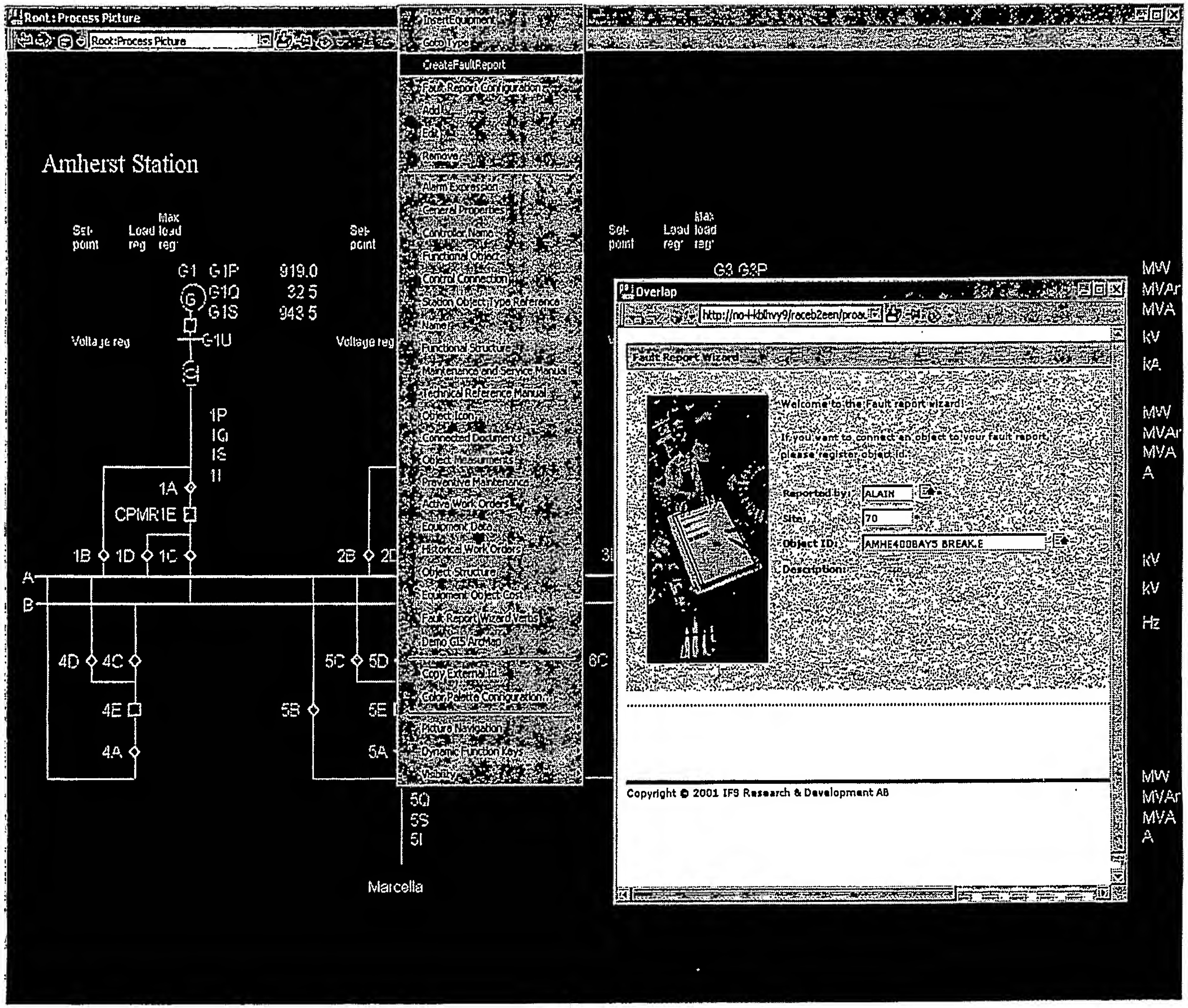


Figure 7

Show geographical map - selected station

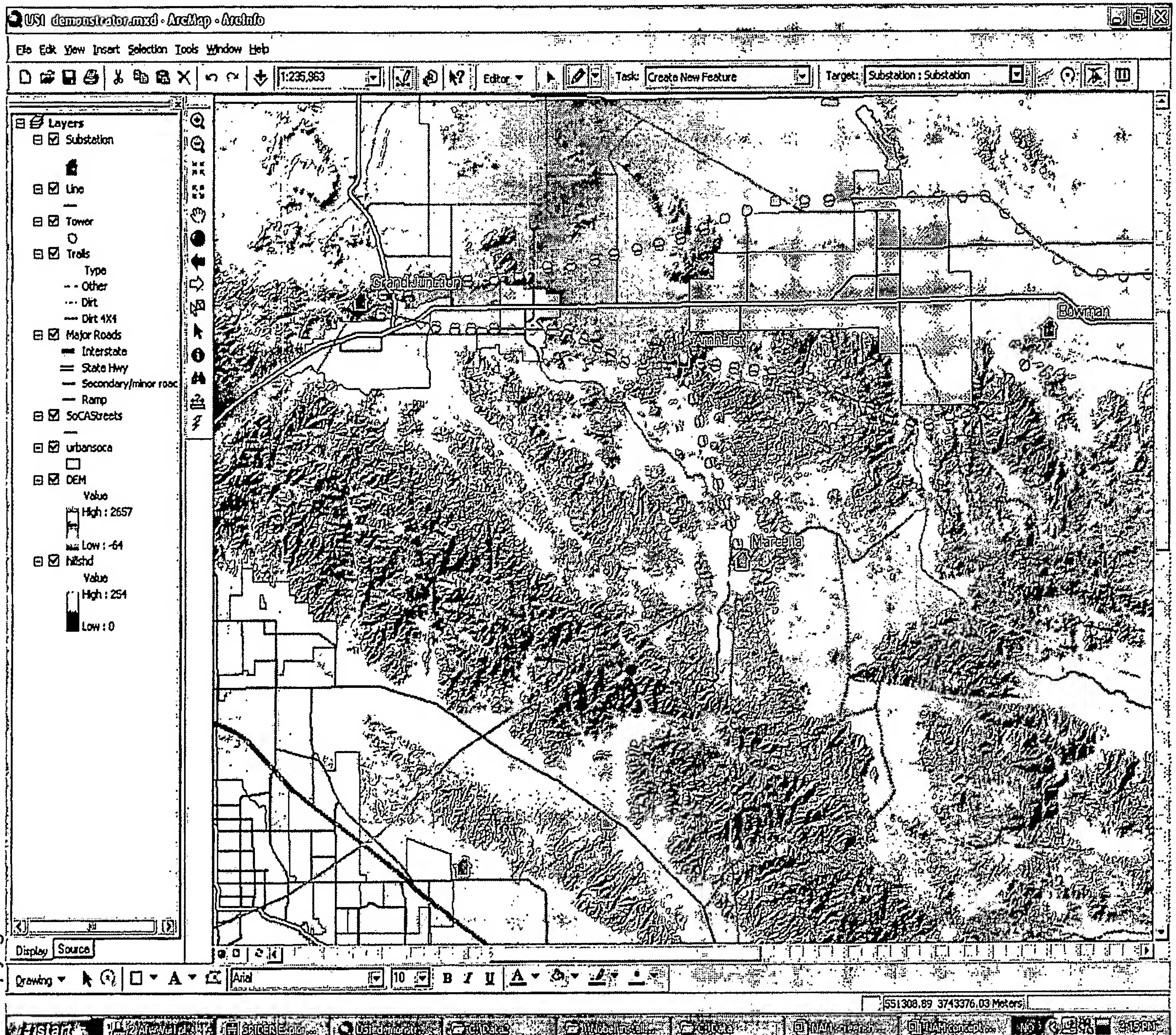


Figure 8

Navigate from map to CMMS (work order)

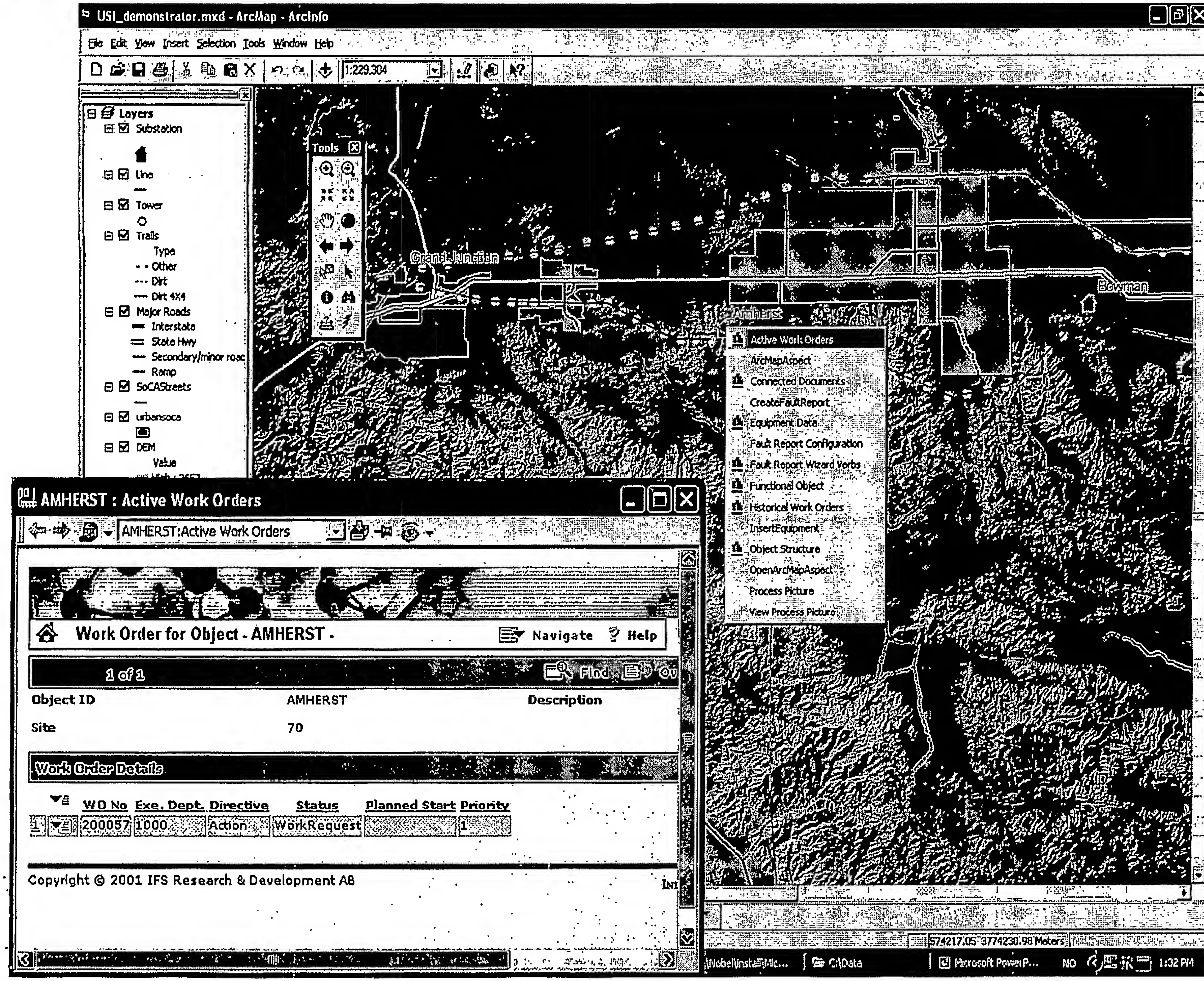


Figure9

Display product documentation (#5)

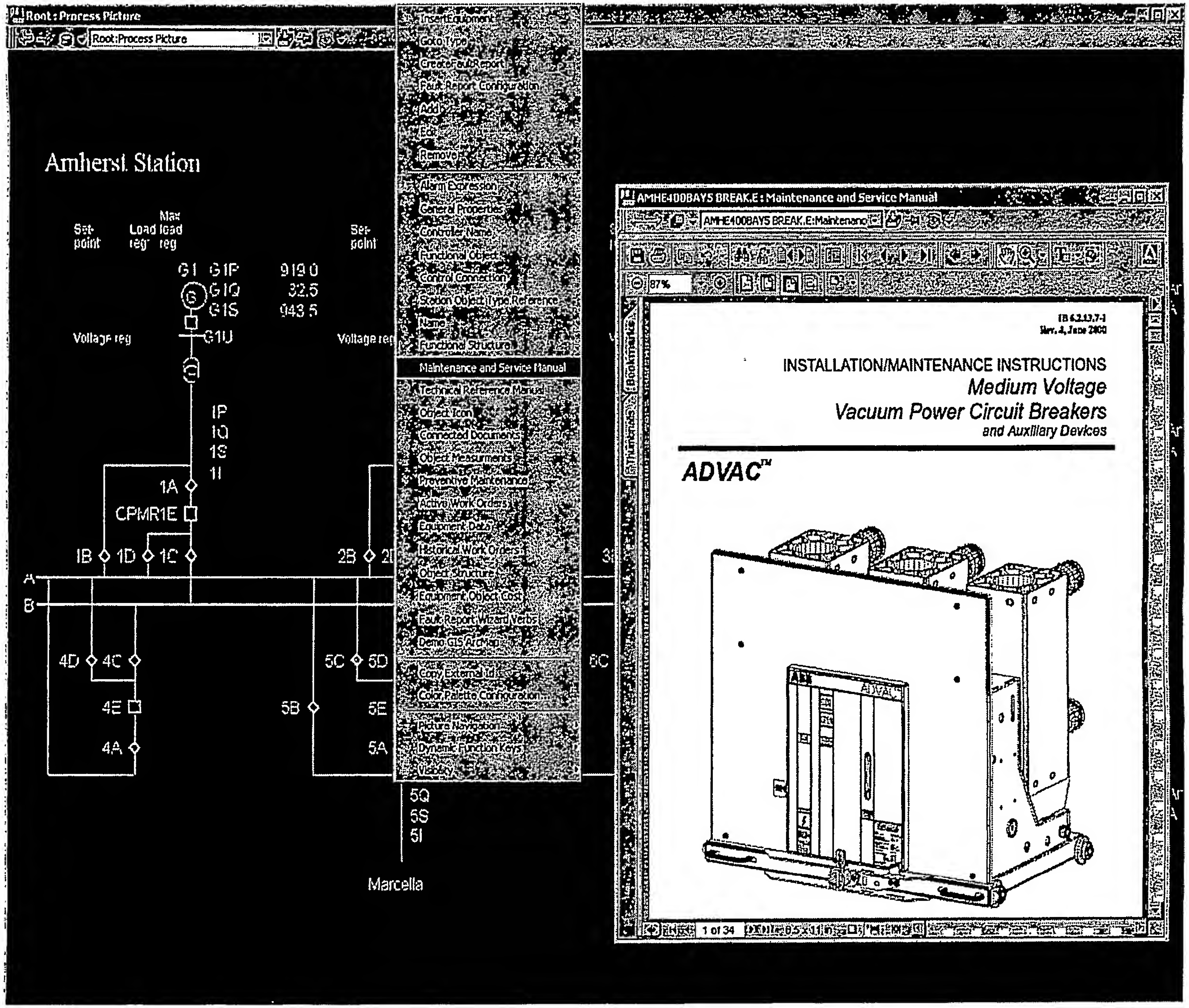


Figure10

ABB

Integrated workplace

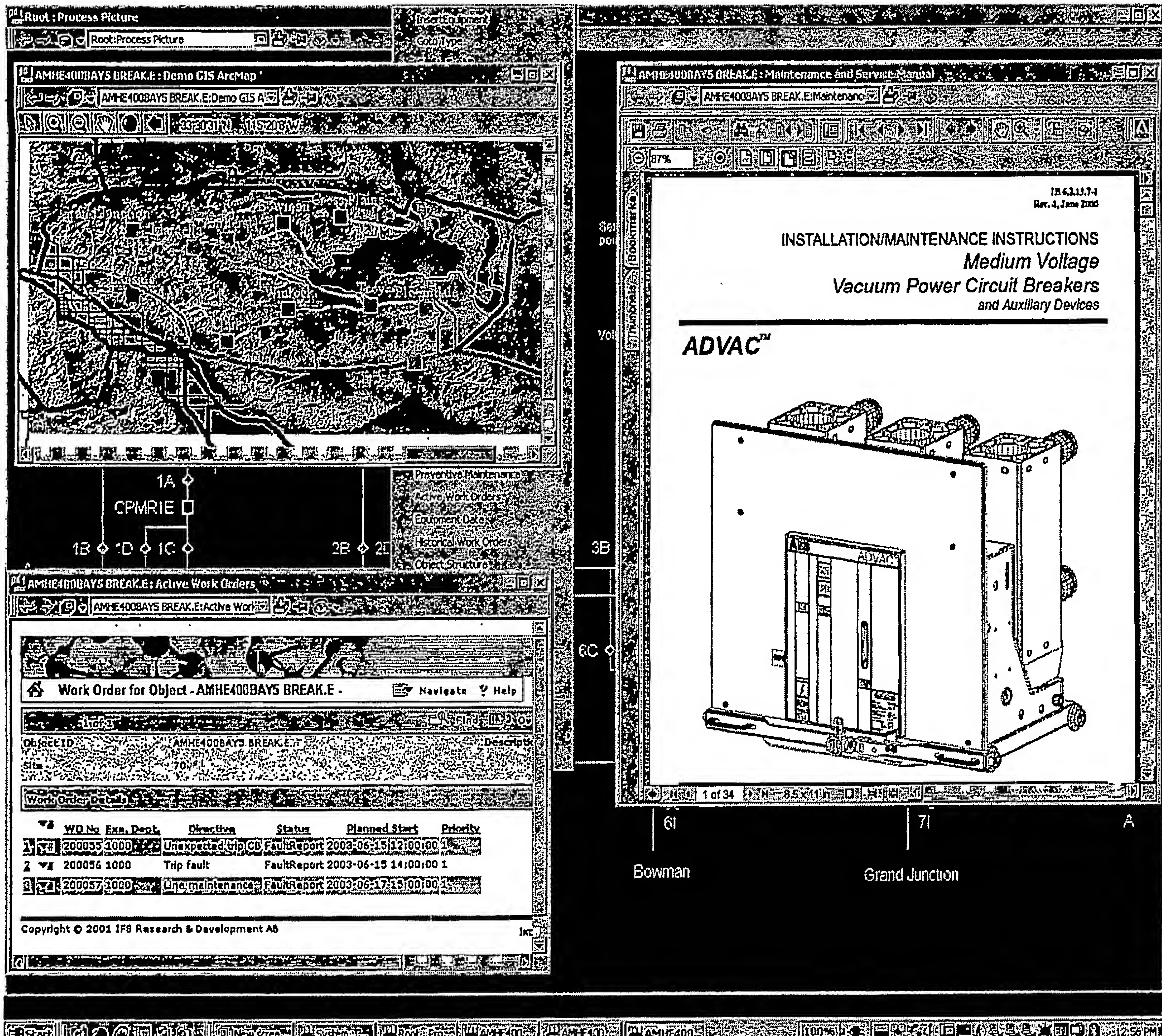


Figure11

ABB

ABB

Data Consistency concept & demo

Data consistency demonstration - concept

- One global / enterprise level data model of the assets (CIM+)
- Attribute consistency – updating attribute values (overlapping)
- Object consistency – adding/deleting objects
- → single data entry
- → One consistent enterprise level : Virtual Asset Register (VAR)

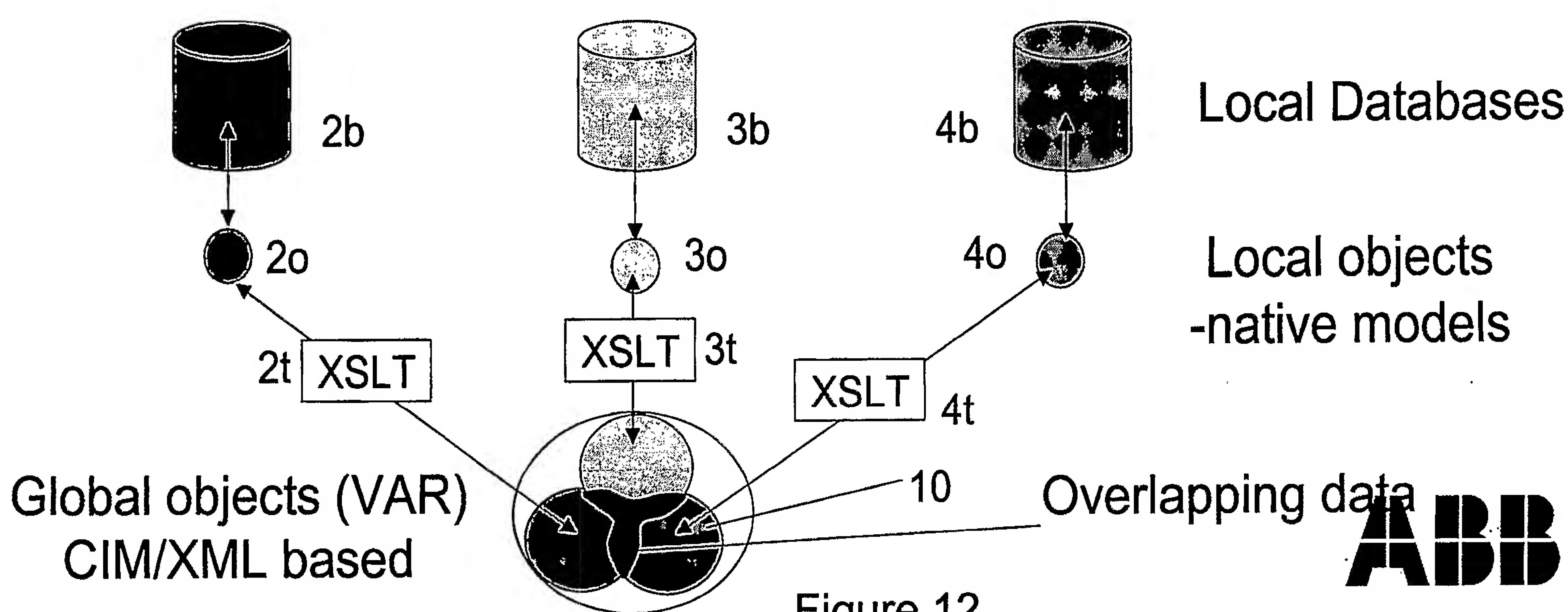


Figure 12

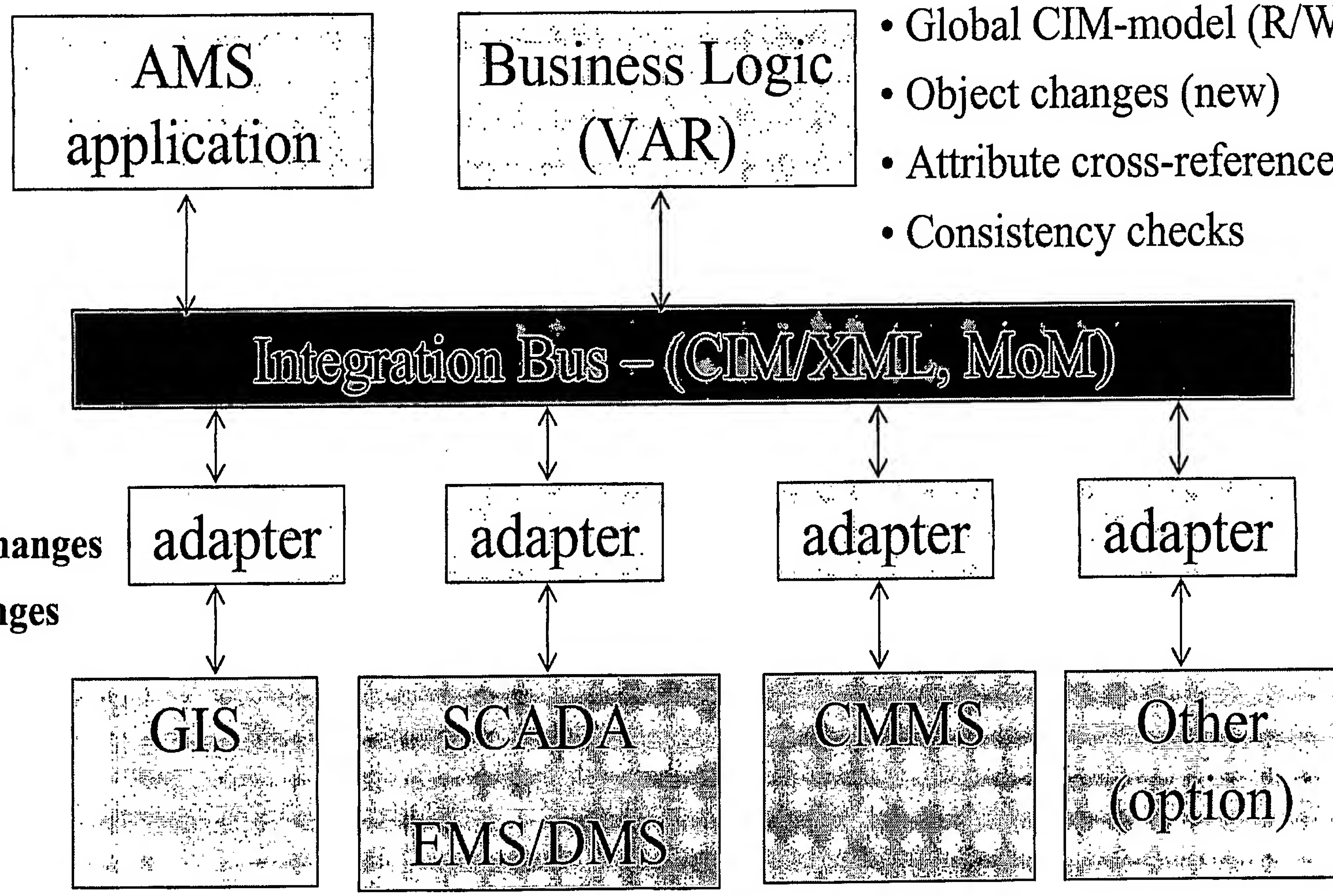
Data consistency / synchronization – basic features

- Add new object (in all relevant systems)
  - Object created in each system based on object templates
  - Connections between systems established automatically
- Delete object (in all relevant systems)
  - Delete defined object in each system
  - Delete object connections (links)
- Access object attributes (all)
  - Select object by identifier (any system)
  - Read out any object property independent of source
- Modify object attribute(s)
  - Select object by identifier (any system)
  - Update attribute in source system (owner)
  - Replicate data to other systems (readers of the data)
  - Maintain object connections (links)

Data consistency - concept

Virtual Asset Register (VAR)

- Global CIM-model (R/W)
- Object changes (new)
- Attribute cross-references
- Consistency checks



- Map XML
- Attribute changes
- Object changes

Figure13

Data consistency demonstrator – Attribute consistency – read / write

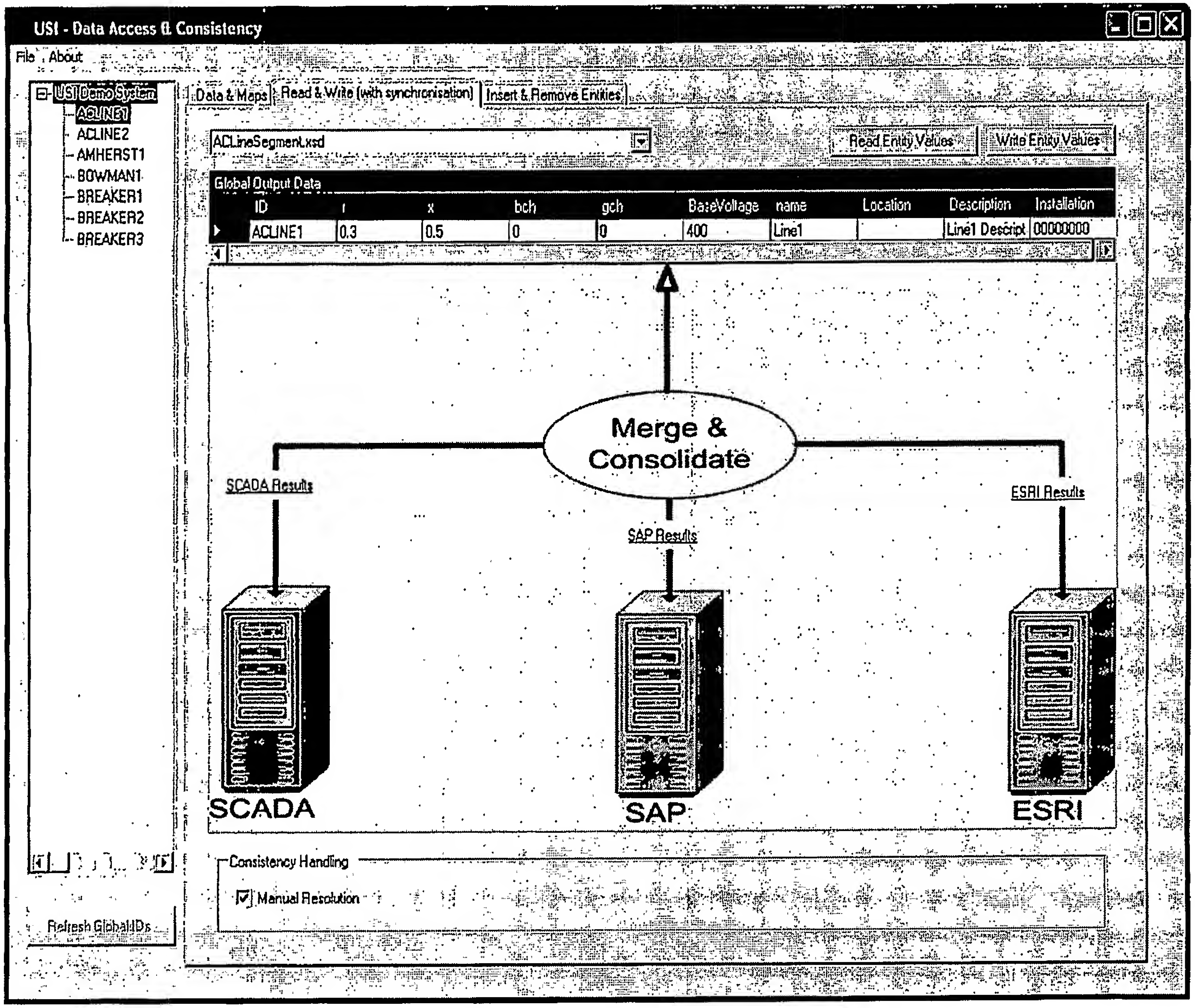


Figure14

Data consistency demonstrator – object model mapping

Mapping between local and global (CIM) data models

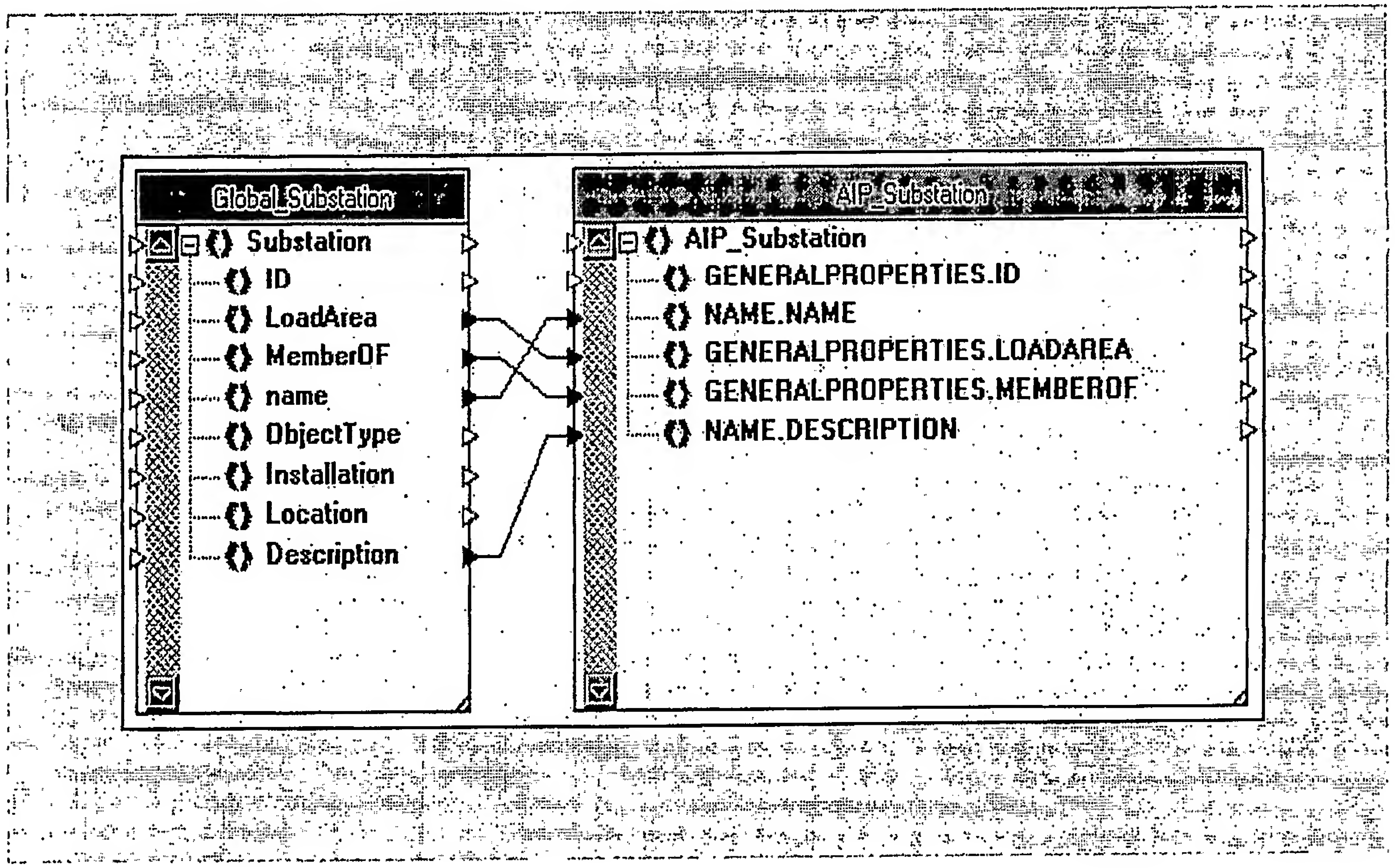


Figure15

Data consistency – Attribute consistency - resolution

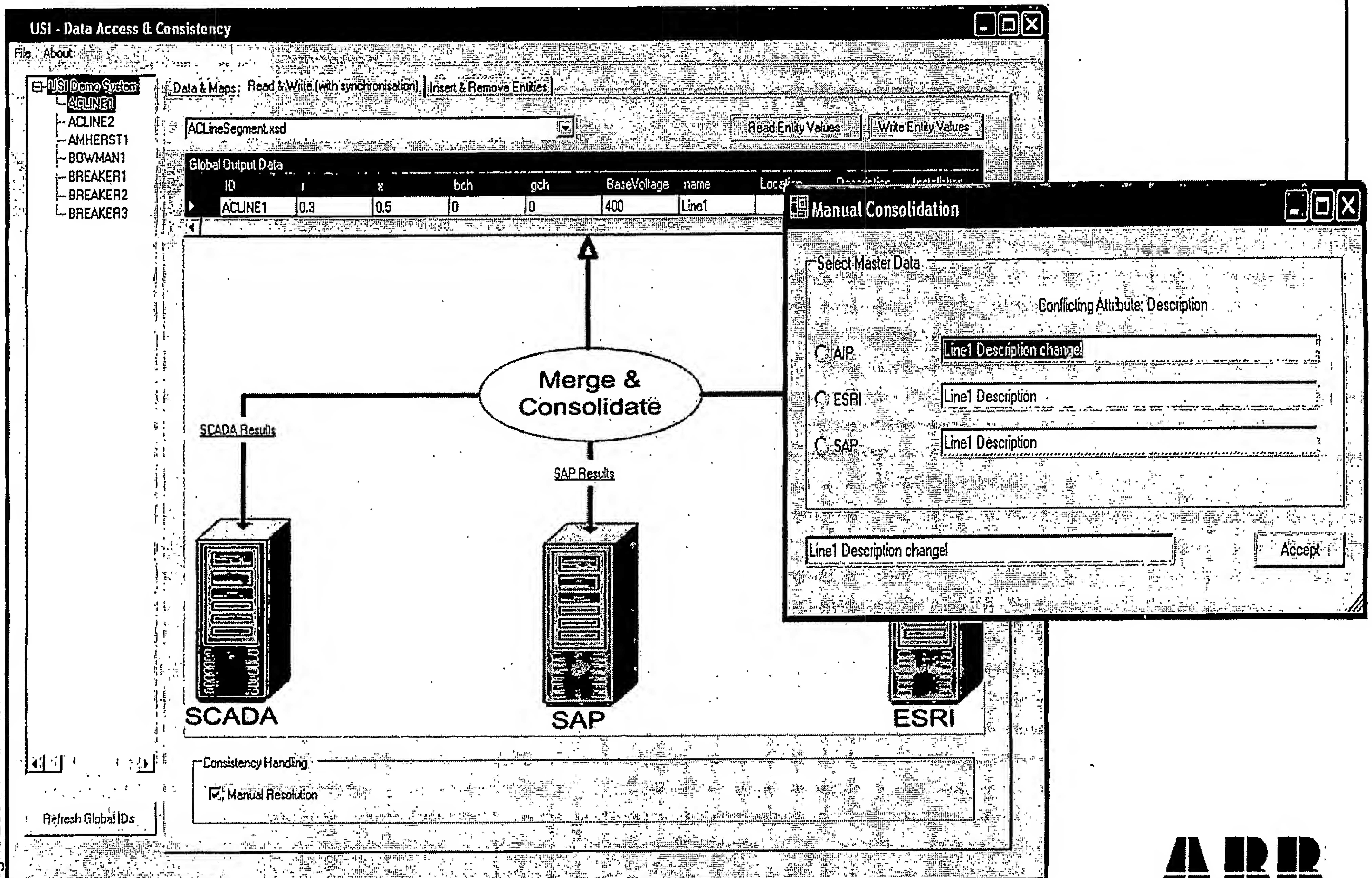


Figure16

Data consistency demonstrator – Insert new object (in ESRI)

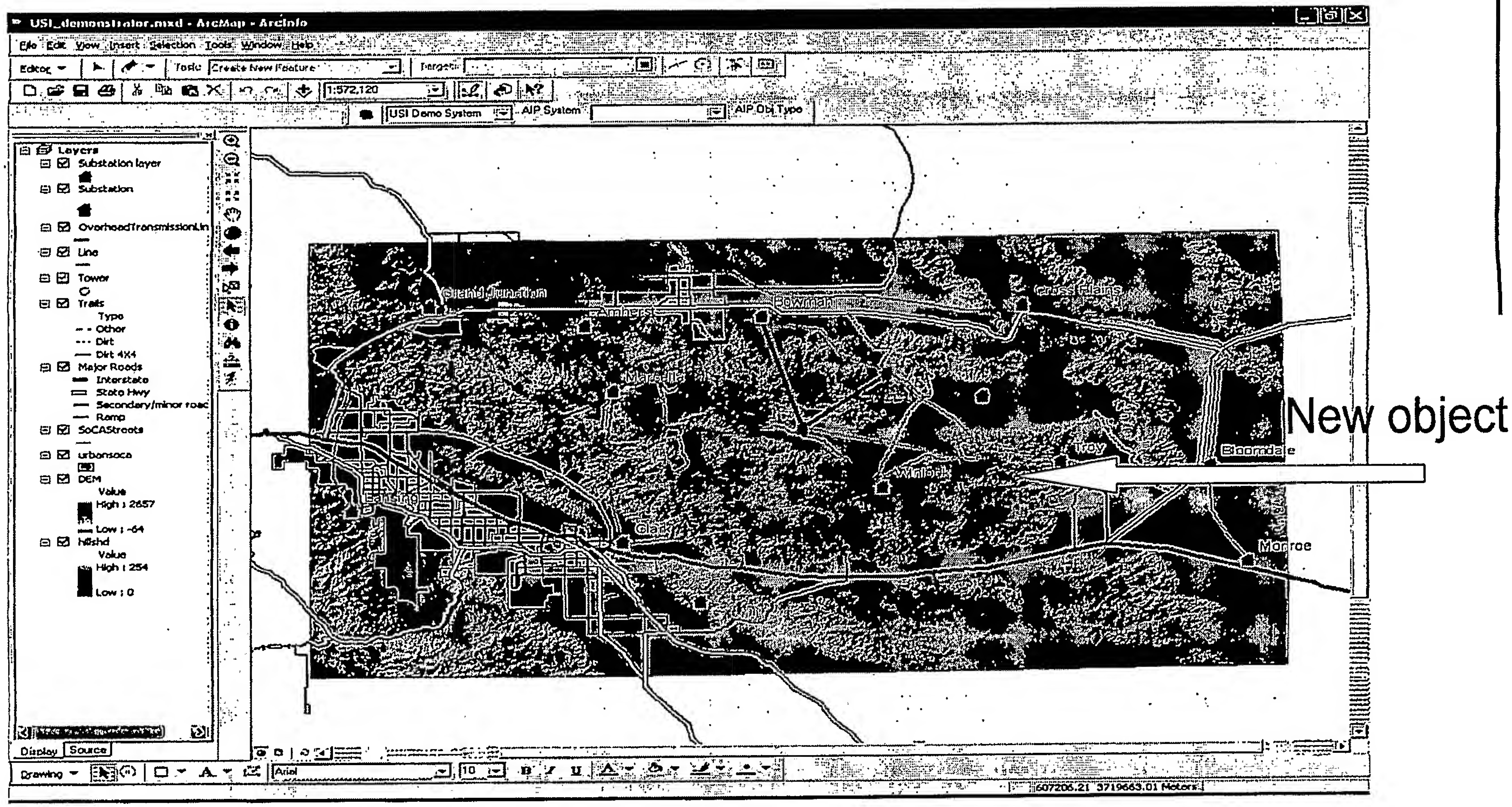


Figure 17

Data consistency - new object recognized

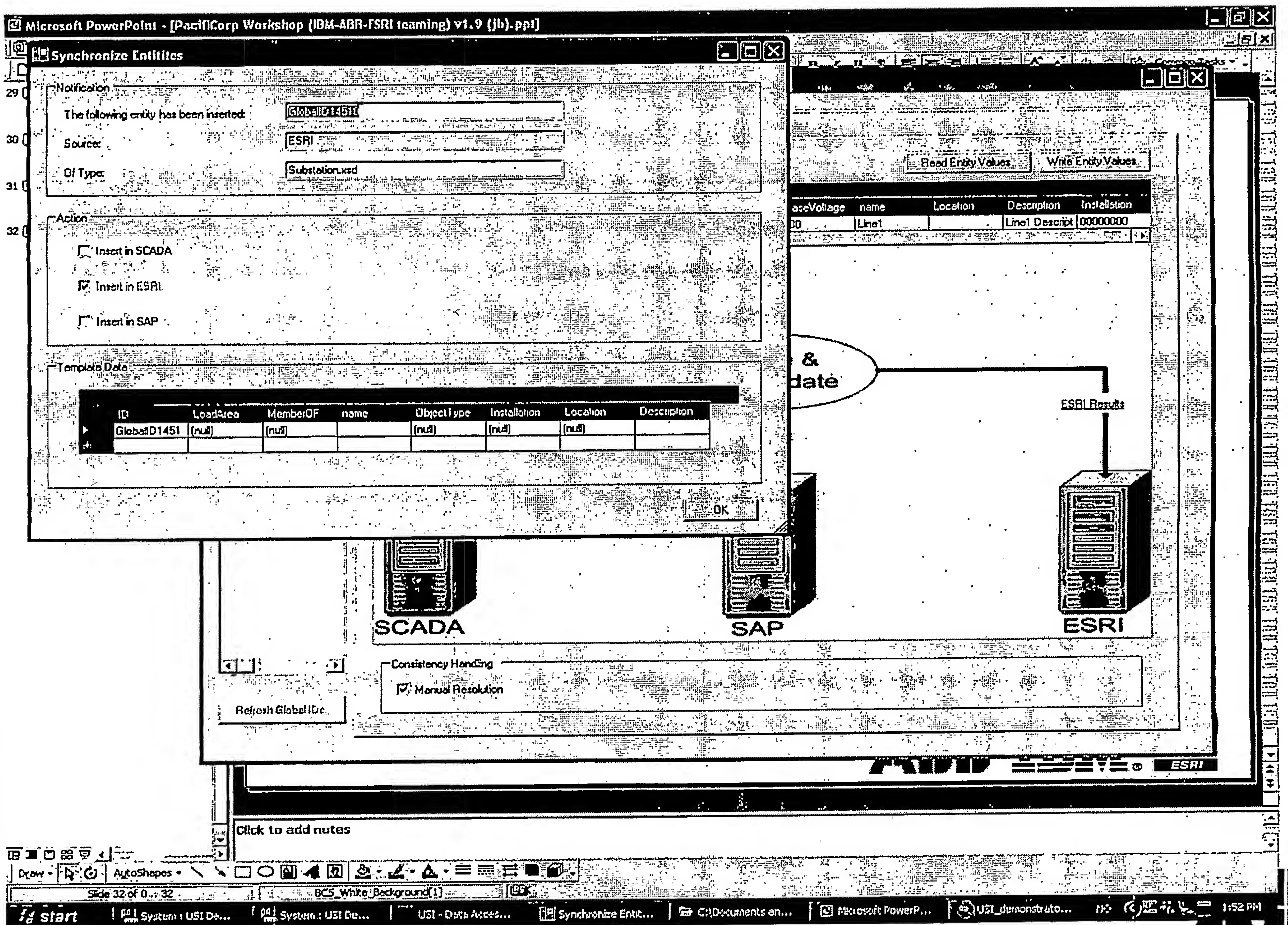


Figure 18

ABB

Backup slides

Demo – fault scenario flowchart

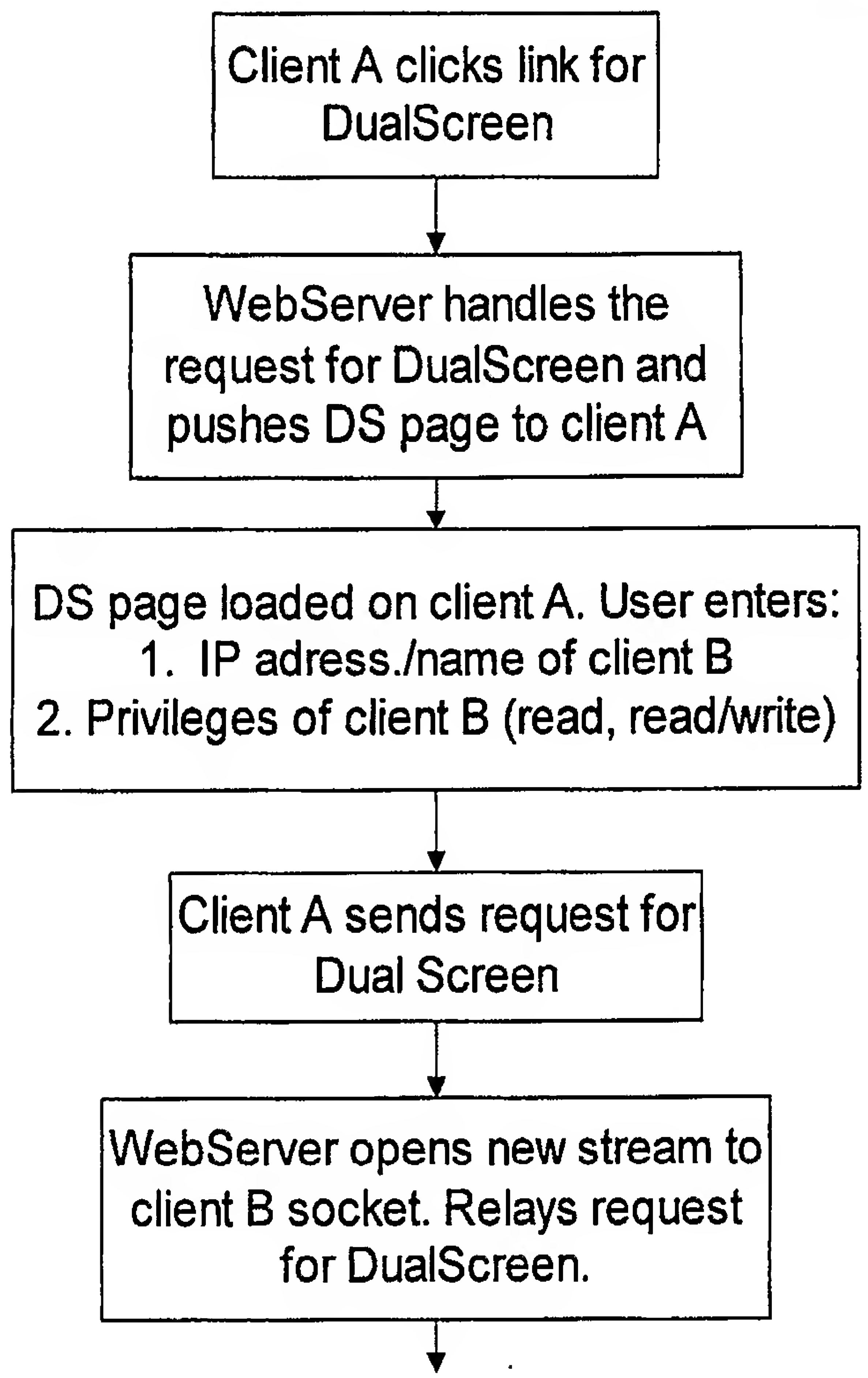


ABB	ABB Corporate Research	CRID30020	
Issued by department: CHCRC.C	Date: 2004-02-17	Lang.: en	Revision: 1.3 / Page: 1 (1)
Doc. name: Phase 2 - Architecture and Design		Status of document:	Draft
Project name: Utility System Integration		Doc. type:	
Creator name: Claus Vetter, Thomas Werner		Distribution:	Reference Group, Project Team

Filename: CRID30020-03-010_P2_Arch_Design; Save date: 2/17/2004 1:00 PM

Architecture and Design Document

Utility System Integration – Phase 2: Data access and consistency

This page is intentionally left blank

Table of Contents

1 INTRODUCTION

Deregulation in the utility industry and the growing use of IT-systems lead to a big increase in IT-integration issues and projects. Utilities require the integration of technical as well as business and commercial IT-systems coming from different suppliers and based on different technology. ABB PTUA is providing several of the IT systems involved in running a utility business, but needs to extend its offering regarding the integration of utility applications and asset management applications.

An umbrella research project [1] was initiated January 2003 with the purpose to support PTUA in developing solutions that encompass integrated utility applications and asset management applications. This umbrella research project has the following goals:

1. To develop concepts and prototypes that deliver user interface navigation between WS500, ESRI, SAP and IFS (Phase 1)
2. <u>To develop concepts and prototypes that deliver improved engineering of data in a utility system landscape, *including online access to data and consistency to SCADA, CMMS and GIS systems. (Phase 2)*</u>
3. To develop concepts and prototypes that deliver asset management applications (Phase 3)

See document [1] for a detailed description of these goals. The umbrella project consists of 3 phases, of which the second is described in this document.

1.1 Purpose of document

This document describes the software architecture for the project "Utility System Integration" – Phase 2 – Data access and consistency.

The document is intended to be a living document and will be updated in an iterative cycle.

2 TECHNICAL OVERVIEW

2.1 Goal of project

This document describes the architecture of phase 2 of the project "Utility Systems Integration" – data access and data consistency. The applications provide the following functionality:

1. Access (read and write) to information stored in the connected IT systems, regardless of the information origin. The data accessed is grouped according to CIM type definitions
2. Detection of data inconsistencies in the underlying target applications, required by synchronization and other services, such as navigation.
3. Synchronization of data sets stored in utility IT systems, according to synchronization rules.

2.2 Generic problem description

A solution that provides data access and consistency to different IT systems encounters the following issues:

1. All participating applications must provide read and write access to their data sets through APIs (database access, OPC access, direct API access).
2. External applications access information where the source system which contains the information is not known a priori, as well as the information which is requested can be stored in more than one system.
3. At the moment of data access and at scheduled intervals, synchronization of the data sets is performed. Therefore, synchronization rules are defined between two or more data sets from different applications.
4. User interaction may be required when data sets are inserted into IT systems. This requires that that IT system is able to provide information about its address namespace.
5. For the definition of type maps, editors from EAI tools (e.g., MS Biztalk), should be used as COTS software components in the engineering tool.

2.3 Expected requirement changes

1. The concept must be flexible enough to include additional applications, such as a Customer Information System (CIS).
2. Through the integration of other systems, additional types serving other needs than operational and maintenance related information, might be required.
3. Additional connectivity to APIs of target applications.

3 FUNCTIONALITY

This chapter describes the functionality provided through phase 2 of the project "Utility Systems Integration" – Data access, and consistency.

Every functionality is motivated through a use case (problem statement), which are followed up in chapter 5 from a technical perspective.

The main concepts referred to in the following chapter are *local objects, attributes, global objects* (relation between *local objects), types for local objects (local types) and global objects (global types)*, and relation between types *(maps)*. *Rules* and *actions* are used to ensure data consistency. For a more detailed description see chapter 4.

3.1 Data access

Data access provides a CIM-compliant, target application independent access to the utility databases, both reading and writing information.

3.1.1 Read / Write from / to target applications

3.1.1.1 Use case

- SCADA user wants to displays (with a specific color) all stations of voltage level 110 kV in which breakers has a scheduled maintenance within the next two months.
- GIS user wants to zoom into a station, show the maintenance schedule for the assets of that station, and change for two breakers the maintenance date.

3.1.1.2 Principal functionality

Analysis and monitoring applications for utilities require information from not only one information source, but combine information out of several source to generate a result, as e.g., use operational information from SCADA, maintenance information from CMMS and combine this information with the network topology in order to optimize maintenance planning for a small region by grouping together maintenance activities.

The added value of such an application is the logic itself, however, looking at the existing infrastructure, main parts of the development go into connectivity to existing utility systems (SCADA, CMMS, Engineering Tool), data extraction and translation.

Data access functionality now introduces a middle layer between the existing utility systems (SCADA, CMMS, GIS, (CIS)) and external applications, which can be e.g., extensions to GIS systems itself for visualization purposes and harmonize the data access both from and to the utility applications.

The main vehicle for that purpose is the introduction of three concepts:

- A type system which is compliant to CIM is exposed for those applications. Applications requiring information can rely on one known model.
- Information from several target applications can be combined in one request. This allows to aggregate information which is related, but stored in different systems, can be combined. This relation is configured during the engineering process.
- Lookup services, which are described in the following chapter.

Access technologies are the following:

- A web service infrastructure is the most platform-neutral access technology
- OPC/DA (DAIS) technologies , APIs
- Message services, with defined commands (REQUEST) and response message definitions for accessing information.

3.1.2 Lookup services

3.1.2.1 Use case

- The GIS user wants to display all breakers within stations in the region “Baden”, and the voltage level of 100 kV. Therefore, the GIS application queries the system for: type=cim:Breaker; Scada region=Baden; Scada voltage level=110
- For the specific breaker with the Scada ID = BA_110_CB45, the users wants to display the next scheduled maintenance. The query towards the system is: return information of type “cim++:Maintenance” for Scada ID = BA_110_CB45

3.1.2.2 Principal functionality

Before an application can access information, it must know the respective "address" of the related information. Two services are available:

- Address lookup (Legacy ID): The *local objects* from the target applications are not directly "visible" to external applications, but the information from the *local objects* is aggregated for external applications. Therefore, the addressing schemes from the target applications can be used in order to request information of a certain type. This functionality is always provided.
- Address lookup (Search queries): *global objects* carry a set of standard attributes with them, e.g., voltage level, region, CIM type. Based on those standard attributes, searches can be executed returning to the external application a result set of matching *global objects* (the return application will receive not the *global object*, but a handle to the *global object*). This functionality is always provided.
- *Optional: Browsing*: Browsing allows, based on a defined topology (e.g., CIM network model), not only to retrieve elements which fit into a specific search query based on certain standard attributes, but also retrieve elements from a relational perspective, as e.g., retrieve all descendants of a substation, etc. Browsing requires that the network topology is engineered together with the *global object* concept described in chapter 4.

3.2 Data consistency

Inconsistency detection and synchronization go hand-in-hand to ensure consistency among the data sets. If an inconsistency in the data sets is detected, e.g., through the violation of a certain *rule*, data synchronization can execute certain *actions* to restore consistency, including involvement of end-users to provide information.

3.2.1 Use case

- If the operational flag for the station BA_110 is changed from "in operation" to "in maintenance" in SCADA, all children of the station in SCADA and CMMS are also set to "in maintenance".
- Operational limits (cim:Winding, cim:XY) for transformer BA_110_T1 are centrally maintained in CMMS system. If one or several attributes change in CMMS, the operational limits must be replicated to SCADA.
- If a line is inserted into GIS, a maintenance element should be inserted into SAP with template parameters, and into the SCADA topology as "cim:line", and an action must

be set that an engineer configures the inserted element in SCADA into the network topology.

3.2.2 Inconsistency detection

3.2.2.1 Principal functionality

Inconsistency detection is a functionality which is not visible to end-users, or to external applications. The main focus lies on the definition and classification of different rules during the engineering phase, which then are applied to the configured data set, either on a cyclic basis, or, if the APIs of the target applications support the feature, triggered through events.

Different rule categories exist, e.g., rules which validate a *local object* as such, rules which compare an attribute against an reference value, or rules which supervise attributes from two or more *local objects*.

Categories allow to define which rules should be applied to which subsets of the engineered solution – e.g., to restrict rules to a certain application category (CMMS only) or application (SAP only), or restrict rules to certain CIM types (cim:PowerSystemResource).

Some of the rules are generated automatically by, e.g., defining a relationship between two attributes of *local objects* in different systems.

If a rule is violated, the concerning *global object* and *local object* are marked, and the information is propagated to the synchronization service to take actions.

3.2.3 Data Synchronization

Data synchronization functionality ensures, based on *actions* that an engineered information set of different target applications is kept consistent.

The functionality is as the data consistency service not directly visible to end-users or external applications. However, synchronization might require user input in the form of re-engineering certain data sets or providing information when the data structure (insert, delete).

The two actions which are provided are

- Synchronization of *local object* attributes
- Synchronization of *local objects*

If the detection of an inconsistency is based on events, the functionality provides near real-time consistent target application databases. If an inconsistency is detected during a poll cycle, a certain time window must be accepted in which data sets are inconsistent.

3.2.3.1 Synchronization of *local object* attributes

3.2.3.1.1 Principal functionality

During the engineering process, type relations (→ see concepts) between *local objects* of one *global object* can be defined. Those relations define how two or more attributes relate to each other. This relationship can be:

- "Ownership" of the attribute (reference value)
- Relation between attributes (e.g., data format conversion (int → string), logical computation (*1000))

As soon as such a relation is defined, it is assumed that a consistency check and synchronization should be executed on this type relation.

3.2.3.2 Synchronization of *local objects*

3.2.3.2.1 Principal functionality

One possible action as a result of a consistency violation can be the modification of a *local object* in target applications, as e.g., the insertion of removal.

Modifications on the network topology (insert, delete) will require manual interaction either through security issues (change permissions) or by the need to supply additional creation information, e.g., where in the network topology a station should be placed. This information cannot be retrieved from a GIS system.

Entry points to synchronization are either changes which are detected on the *local object* lists of the source systems, or through a defined functionality, which is available to a user. In either way, the trigger (user starts action, or change is detected) can spin-off a workflow of sequential actions which are processed in order to restore consistency in the systems.

Examples for workflows are, e.g., inserting new *local objects* in systems, or removing a *local object* from the databases. In order to keep the synchronization service itself consistent, mapping information needs to be added or removed at the end of the workflow.

Inserting a new *local object* requires input to the *local objects* which are created. This input can be taken from default templates (for each system), or be provided by an external source beforehand, e.g., and engineering application, or during the creation process, e.g., through additional user input. If the inserted type shares attributes between the systems, those attributes can also be used as input parameters for *local object* instance created in other systems.

The deletion of a *local object* requires authority to remove, or change the status, of a *local object* in the source systems, and to remove its mapping information from the consistency service.

The executed sequence must be configurable, being able to adapt the creation process according to the processes as they are applied in the utilities today manually. Configuration can include, e.g., which systems are called in which order during creation; at which point's user authorization is necessary; which information can be taken from a template or must be provided by a user during the engineering process etc.

4 CONCEPTS FOR INFORMATION INTEGRATION IN HETEROGENEOUS UTILITY SYSTEMS

The main concepts phase 2 builds upon are *local objects, attributes, global objects (*relation between *local objects), types for local objects (local types) and global object (global types)*, and relation between types *(maps)*. *Rules* and *actions* are used to ensure data consistency and synchronization.

The realization of the use cases with the underlying concepts and technologies is described in chapter 5.

4.1 Local object

A *local object* is a representation of a data set in one of the applications, e.g., a equipment data set in CMMS, or a breaker instance in a SCADA, or a substation element in a GIS.

A *local object* can always be uniquely addressed through a set of identifiers.

4.2 Global object

A *global object* describes a relationship between *local objects* from different applications. In most cases, a *global object* corresponds to a physical asset in the electrical network, e.g., breaker, transformer, bay, or substation, which has different representations (local object) in the connected IT systems

A *global object* can be assigned to CIM instance identifiers (and access paths: relation to other *global objects*).

4.3 Attribute

Attributes are the values associated with a data set in the application.

Local objects dynamically expose attributes according to their local types, as well as *global objects* dynamically expose attributes according to their global types.

Attributes therefore can be accessed from external applications, such as asset management. Attributes are directly accessed from the corresponding target applications.

4.4 Types (Local, global) and maps

Local types describe the attribute sets which the target applications can expose on their *local objects*. In most cases, local types are proprietary to the respective target applications.

A local type is always bound to a *local object*, therefore, the local types also contain a relative address to the bound *local object* in order access the attributes.

Global types are derived from CIM (and other standards) and describe the attributes which are exposed to external applications. Global types do not carry a relative address for each attributes. A *global object* can carry several global types.

Type maps define relationships between local-global types.

4.5 Data view, data access functions

Data views define functionality on the global types, e.g., GetMaintenanceOrders, AdaptOperationalLimits. Those functions mostly work on several types exposed, or even on several *global objects*.

4.6 Rules and Actions

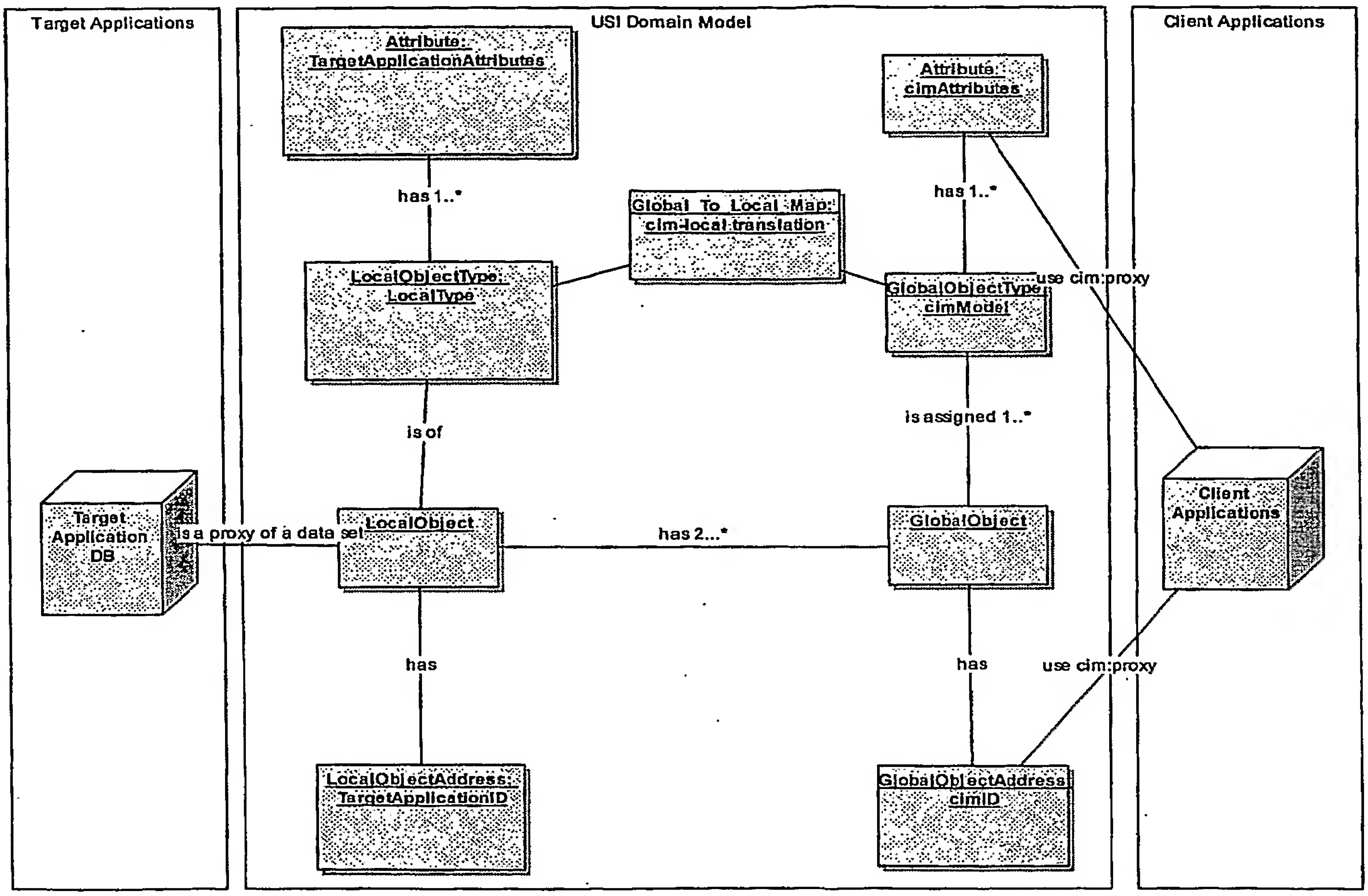
Rules are used to detect consistency violations in the data sets of the applications. Rules are configured during engineering either explicitly, or implicitly e.g., by type maps.

As a result of a consistency violation, a sequence of actions can be executed in order to restore the synchronization. These actions can be the modification of the data topology (insert, delete *local objects*), or synchronizations of *local object* attributes. Actions can be chained, offering the possibility to not only insert one element in a system, but to perform an insertion of several elements (e.g., substation – one element needs to be inserted in GIS, but different elements need to be inserted in SCADA). These available composite actions for each system can be configured.

Actions may also require user confirmation and input.

4.7 Integration concept – Overview picture

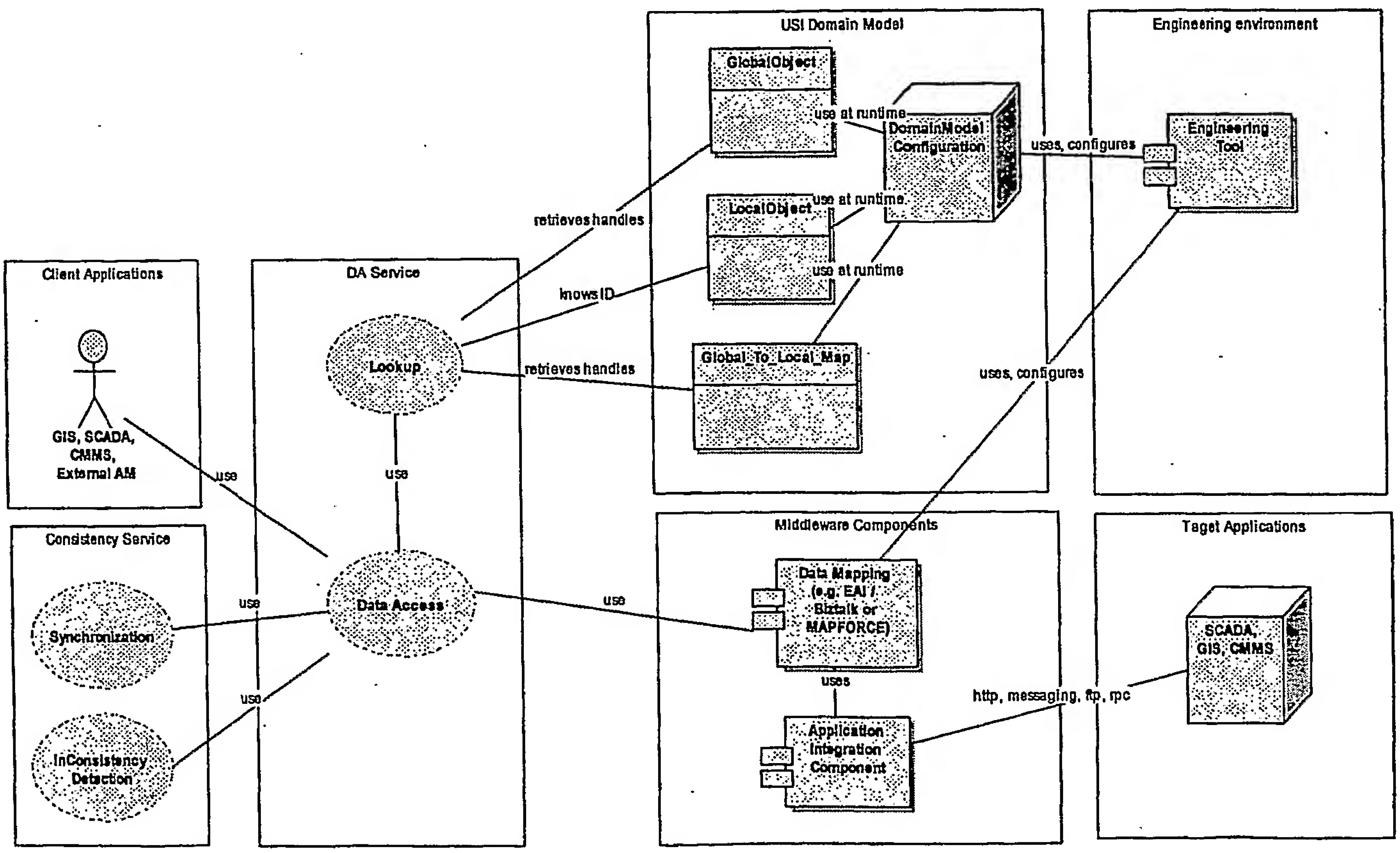
The following picture illustrates the concepts and their relationships among each other.



5 ARCHITECTURE OVERVIEW

5.1 Software architecture overview

The following figure shows the software architecture from a high level perspective, including the usage of middleware products.



The following functionality is required to realize data access and data synchronization. The functionality also defines the main packages for phase 2

5.1.1 Target applications

The phase 2 functionality uses the API's which are exposed by the different target applications to establish at runtime an asynchronous communication link.

The main functionality which needs to be provided through API's is read and write of attributes, as well as insertion and deletion of *local objects*.

API's are provided with different technologies. For all cases, *direct* access the databases of the target applications are not allowed because it would cause data integrity issues. The access instead is encapsulated through:

- SCADA:
  - Read/write: OPC/DA or API
  - Insert/delete: API
- ESRI GIS
  - Read/write: MS COM Automation model, .NET in v8
  - Insert/delete: MS COM Automation model, .NET in v8
- SAP CMMS
  - Read/write: SAP .net connector
  - Insert/delete: SAP.net connector
- IFS CMMS
  - Read/write: API
  - Insert/delete: API

5.1.2 Application Integration Components

Data access and synchronization uses so-called Application Integration Components to communicate with the target applications to execute the two basic functions

- Read/write attributes (data access)
- Insert/delete elements (data synchronization, "actions")

AICs provide towards the middleware services a consistent interface, and encapsulate the specifics of communicating to a target application.

5.1.3 Middleware components

Microsoft's Biztalk data mapping functionality will be used as technology for a prototyping solution to prove the functionality of phase 2, in conjunction with AICs for the different target systems.

However, through defined interfaces from the core domain model, also other middleware products, such as IBM WebSphere or TibCo, can be used in the architecture.

5.1.4 USI Domain Model and Services

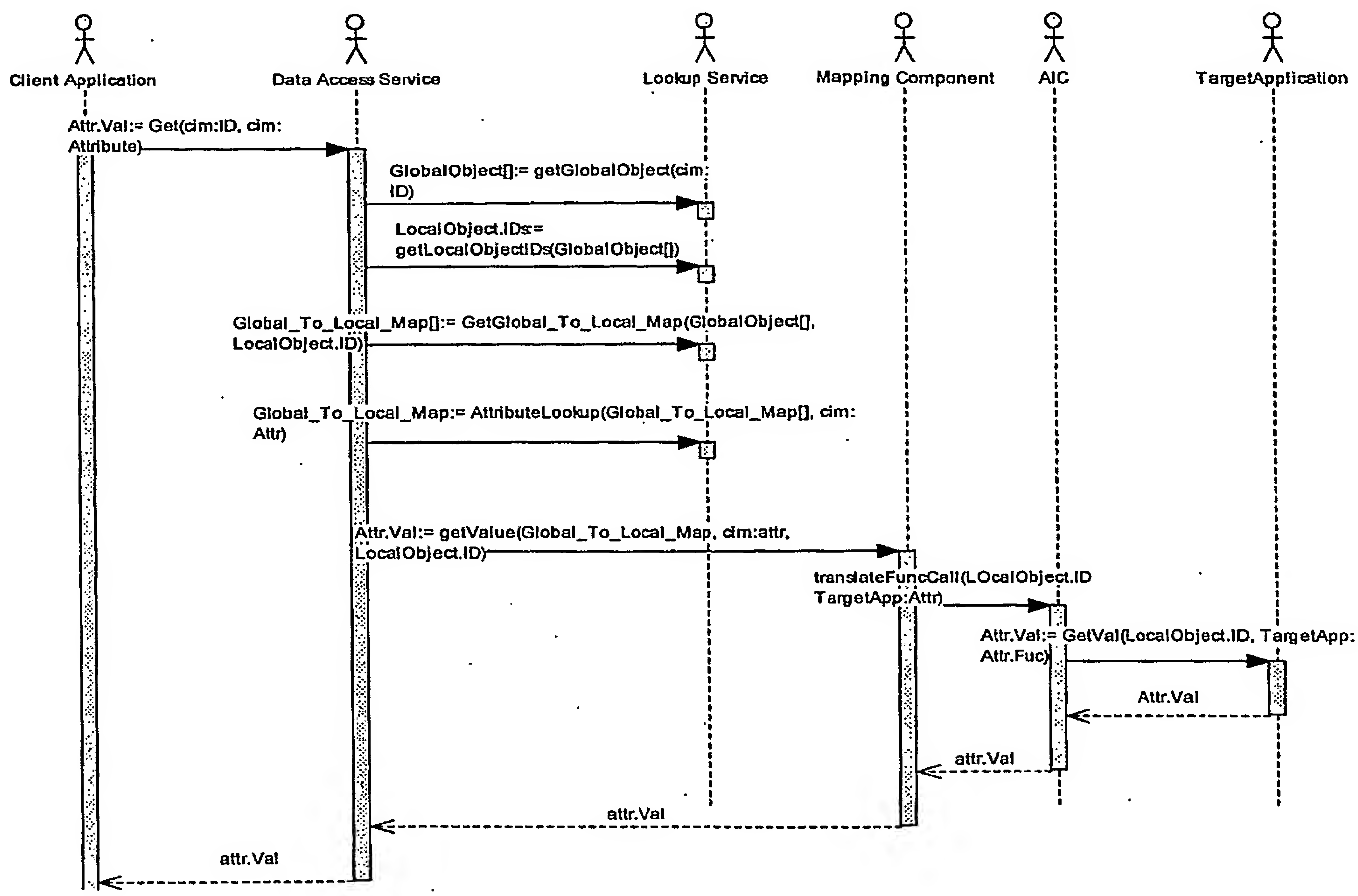
The USI domain model provides is the runtime representation of the configuration database. The domain model (*global objects*) also delegates data acquisition to the middleware services.

Services represent the visible layer for applications to access the USI functionality, such as lookup and data access. The services consistency and synchronization can be configured from outside applications.

The services can be provided with different access technologies. For distributed applications, the most simple to handle are web services (SOAP), but also a messaging protocol handling CIM or DAIS commands is possible.

5.2 Realization of selected use cases

For lookup and data access (without consistency checks and synchronizations), the following sequence diagram applies:



5.2.1 Data access

- SCADA user wants to displays (with a specific color) all stations of voltage level 110 kV in which breakers has a scheduled maintenance within the next two months.
- GIS user wants to zoom into a station, show the maintenance schedule for the assets of that station, and change for two breakers the maintenance date.

The SCADA add-on has a *global object* to the USI lookup and data access web services. It queries the lookup service for a collection of assets (cim-type: breaker), where the attribute of

the voltage level is 110 kV (cim-type: basicInfo), and where the attribute for maintenance interval is scheduled for the next two months (cim++-type: maintence).

The lookup service selects all *global objects* which carry both a cim-type: breaker, and a cim-type: maintenance, and a cim-type: basicInfo.

Of this collection, only those *global objects* are returned to the SCADA add-on in which the attributes for voltage level and maintenance interval date do match. Therefore, the lookup service automatically uses the data access services and compares for the whole collection if the attributes for voltage level on cim-type: basicInfo match to 110 and the attribute for maintenance data on cim-type: maintenance does match to the specified date.

The SCADA add-on will retrieve server handles of *global objects* which fulfill the lookup request, and which can be used for further operations.

For one of the breakers, the operator modifies the maintenance date. The GIS add-on connects to the data access service, by using the corresponding server handle, and provides the attribute "maintenance date" with the modified value.

The data access service retrieves the *global object* for the specified handle. If a cim-type "maintenance" is defined on the *global object*, then the data access service initiates a write-operation, using the provided value, the cim-type: maintenance, and the selection that all connected target systems should be updated.

The USI domain model now retrieves all configured maps for the cim-type maintenance. There is only one map returned, which defines a mapping to a local CMMS type: maintenance. Here, the attributes for maintenance date are connected.

The map is now submitted, along with the attribute, to Biztalk mapping service. The mapping service performs a string-date conversion on the attribute. Since the target type is a CMMS element, the Biztalk mapping service calls the target application adapter, by supplying the CMMS *local object* ID from the *global object*, and the modified attribute.

The adapter now connects to the CMMS system, locates the maintenance report (*local object* ID) and in another call updates the attribute.

5.2.2 Data consistency

As an example, a consistency check is performed as soon as an external application accesses the domain model (read).

Before the read request is delegated to the EAI components, the domain model performs a consistency check on the *global object* and passes the *global object* to the consistency service. Here, all rules which have been configured (*local object*, system, specific attributes) are retrieved and sequentially applied (rule validation). If one rule fails, this is logged in the status history of the *global object* or the *local object*, and an event to the sync service is sent.

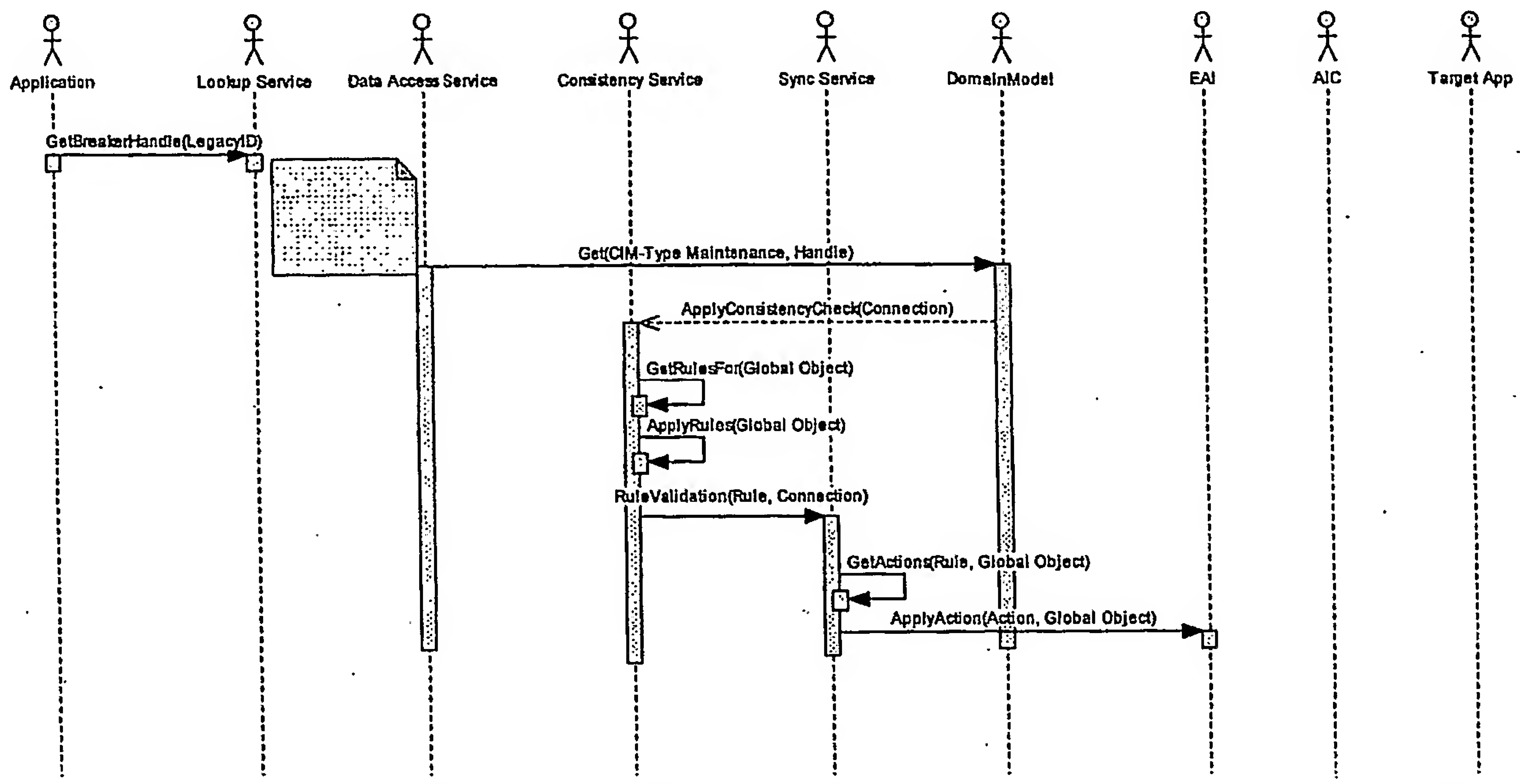
5.2.2.1 Notification mechanisms

Consistency relies on certain notification mechanisms which allow triggering the service. Three notification mechanisms can be differentiated. Implementation of those is depending on the functionality of the target application APIs.

- Poll – The poll mechanism is built in and works on continuously running through the USI configuration (*local object*s, *global object*s) and applying corresponding rules. Polling accepts a duration in which an inconsistency between the data sets can occur. The duration, however, can be guided through the assignment of priorities, meaning that some objects needs to be checked for consistency more often than others.
- Event-triggered – If a client application requests e.g. attributes from a known *local object* identifier, a consistency may be performed before returning information to the client application in order to ensure consistency on the requested information.
- Push – If supported by the target application APIs, those applications may fire an event if they detect a change in their configured address space, e.g., if a *local object* is being modified, or if objects are inserted or deleted. The consistency service then can react on a push event and execute actions to re-establish consistency.

5.2.3 Data synchronization

The following diagram shows the principle behavior of the data synchronization service



Synchronization is triggered out of the consistency service as soon as a rule is violated. The consistency service calls the sync service alongside with the rule that was violated, and the *global object* on which the rule has been applied.

The synchronization service queries a list of available actions for the (rule, *global object*) pair. The following two principle actions are considered in the first release:

- Update attributes
- Insert or remove *local objects* in a system with template parameters (w/o user interaction)

Specifically, for the following use case, the sequence diagram is illustrated in detail:

- If a line is inserted into GIS (Rule), a maintenance element should be inserted into SAP with template parameters (action), and into the SCADA topology as “cim:line” (action), and an action must be set that an engineer configures the inserted element in SCADA (test system) into the network topology (action).

6 PACKAGE DESCRIPTION

Phase 2 consists of the following packages

- Domain Model and service functionality
  - Lookup and Query services
  - Data Access
  - Data consistency
    - Rules: *local object*; Attribute of *local object*; Relation between attributes of n *local objects*
  - Synchronization
    - Actions: Synchronize attribute (overwrite), insert or remove one element from a template
  - Deployment
- Data and type models
  - CIM class library and CIM schemas (global types)
  - Definition of access functions
  - Definition of local types
- Application Integration Components (IFS, SAP, SCADA via OPC, ESRI)
  - Engineering Adapters
  - Runtime Adapters
- Engineering tool extensions to P1
- Databases and database tables extensions to P1
- *Optional*
  - *Browsing in network topology and modification of network topology*
  - *Additional Rules and actions for consistency and synchronization*

6.1 Package: Domain model and service functionality

The package defines the class library necessary to realize the concepts of *local objects*, *global objects*, types, and maps.

The package also defines the services which work with the *local objects* and *global objects*: consistency, synchronization, and data access.

Within the data access service, an EAI data mapping functionality is embedded which works with local and global types (schemata) in order to provide a data mapping.

6.2 Package: Data and type models

The package defines the access functionality for external applications. This includes the introduction of a CIM compliant .net class library which can be used by .net enabled applications to request information and modify the CIM model.

CIM types are derived from the CIM rose model in order to introduce global types, as well as schemas for use in data mapping tools.

Local types with relative attribute addressing are defined which are assigned to *local objects*.

Data access functions are introduced and defined which operate on either a series of *global objects*, or a collection of CIM instance classes in order to provide functionality which are common to many applications, as e.g., provide maintenance functions, or functions which allow to modify operational limits.

6.3 Application integration components

The package designs and develops the application integration components necessary to interact with the different target applications and to define a common interface set towards the domain model in order to encapsulate the application specific API.

Both for engineering and runtime application adapters are developed.

Engineering adapters are able to parse an application export file and generate a set of *local objects*.

Runtime adapters are able to read and write attributes, and to modify the address space of the target application (create, modify and delete *local objects*).

6.4 Engineering tool extensions

For phase 2, additional mapping information is required in order to realize the functionality. The engineering tool will be extended with the corresponding wizards to

- Assign local and global types to *local objects* and *global objects*
- Define attribute relations between local and global types, and between local types of different systems (synchronization). Here, data mapping wizards from EAI tools may be employed.
- Define consistency and synchronization rules for *local objects* and *global objects*
- <other?>

6.5 Database table extensions

This package defines the database tables and access functions in order to handle phase 2 concepts.

6.6 Browsing

Optional:

Requires *global objects* (paths) are engineered with a CIM-XML network topology.

7 FUNCTIONAL DESCRIPTION

The functional description specifies in detail the packages introduced in chapter 6, the components that reside in the packages, their interfaces, interactions and dependencies. It is prepared for Gate 3

7.1 Package <Package Name>

Help: *Describe per package the components. For AIP, describe for an ASO level the interfaces, and provide in detail how the Aspect is built up out of Data, View and Verb ASOs. Are other, non-ASO components involved?*
Describe interfaces and properties, relationships between components, their behaviour (sequence diagrams). Which properties of the components are persistent, is the component running as a Server Aspect?
Provide an overview graphics of the package in UML notation. What are the relationships to other components residing in other packages?

7.1.1 Component: <Component Name>

Help: *Per component, provide a class diagram and a sequence diagram that outlines how the component is activated within the system context.*
Which interfaces does the component implement and what is the signature of the interfaces? For AIP, which properties are provided as OPC properties, and which of the properties are persistent?
On which Object Types can the Aspect been introduced?

7.1.1.1 Description

7.1.1.2 Behaviour

7.1.1.3 Interfaces and Methods

7.1.1.4 Properties

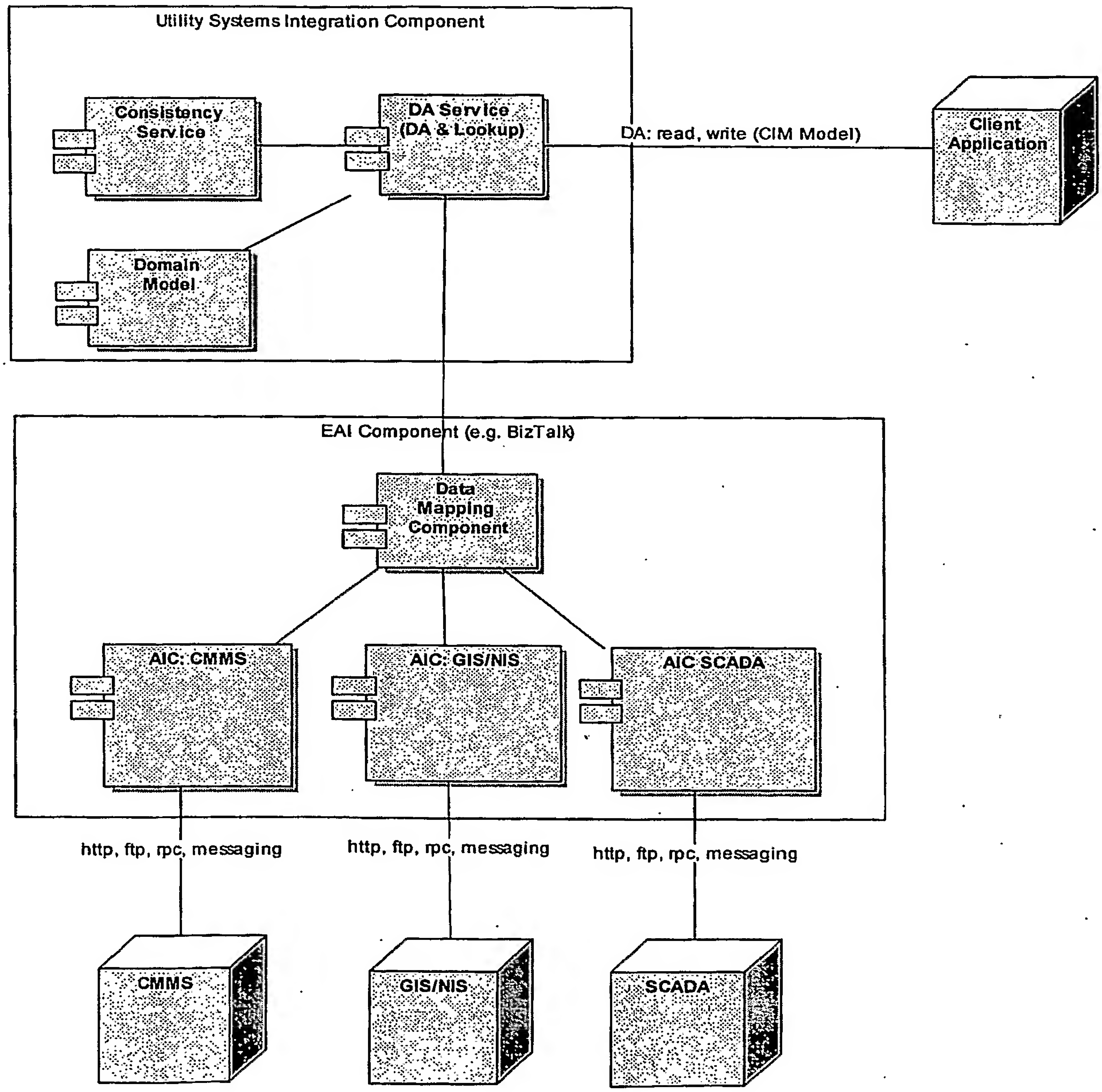
8 SYSTEM DEPLOYMENT

8.1 Description

The following picture outlines the system deployment, focusing on a “centralized” solution. An alternative is described in chapter 10, alongside with advantages and disadvantages.

The centralized solution is deployed in the following way:

- Client applications requesting services from the USI Domain Model communicate via dedicated protocols (web services, messaging) with the domain model services (lookup, data access). Client applications may be deployed on the same server machines as the domain model services.
- The core domain model (configured local and *global object*s, loaded at runtime), through which lookup and data access services request information, as well as consistency and synchronization services are deployed on one or several servers. The architecture is designed to allow distributing the models on several machines utilizing .net.
- The EAI data mapping component is deployed on the same server as the core domain model services for a small system setup, and can be deployed on an additional server for large system configurations.
- Target application wrapping components, which are utilized by the EAI components, will in most cases be deployed on the servers on which the EAI component is running. For target applications which will exchange a lot of information (SCADA), the wrapping component can also be deployed on an additional machine. The communication towards EAI is secured through protocols like messaging or web services.



Utility Systems Integration Component
Consistency Service
DA Service (DA & Lookup)
Domain Model
DA: read, write (CIM Model)
Client Application
EAI Component (e.g. BizTalk)
Data Mapping Component
AIC: CMMS
AIC: GIS/NIS
AIC SCADA
http, ftp, rpc, messaging
http, ftp, rpc, messaging
http, ftp, rpc, messaging
CMMS
GIS/NIS
SCADA

8.2 Discussion

- Wrapping components for target applications can be kept relatively "thin" – the basic functionality which needs to be implemented is some generic read/write attributes, and some insert/remove *local objects*. If supported by the target application, the application adapter can also propagate events on changes in the target applications' configured address space (inserting or removing *local objects*).
- Only one EAI component will be deployed (which might be distributed on several machines for load balancing), thus reducing licensing costs.
- The configured domain model also servers as central lookup services to resolve any requests from client applications towards a standardized access. The *global objects* can be assigned a common address space which assigns a unique naming to the *global objects* (e.g., CIM instance naming), but it is not necessary since *global objects* can also be identified through attributes which are carried by *local objects*, e.g., type information, *local object* ID information etc.
- Since all lookup and data requests are routed through the same central components, those need to be designed carefully to allow e.g., for a distribution to several servers (load balancing).

9 DATA VIEW

Prepared for GATE 3

9.1 Database tables

10 ADDITIONAL INFORMATION

10.1 Project Risks

Project risks are identified and updated continuously in the project plan for the project.

10.2 Alternative solutions

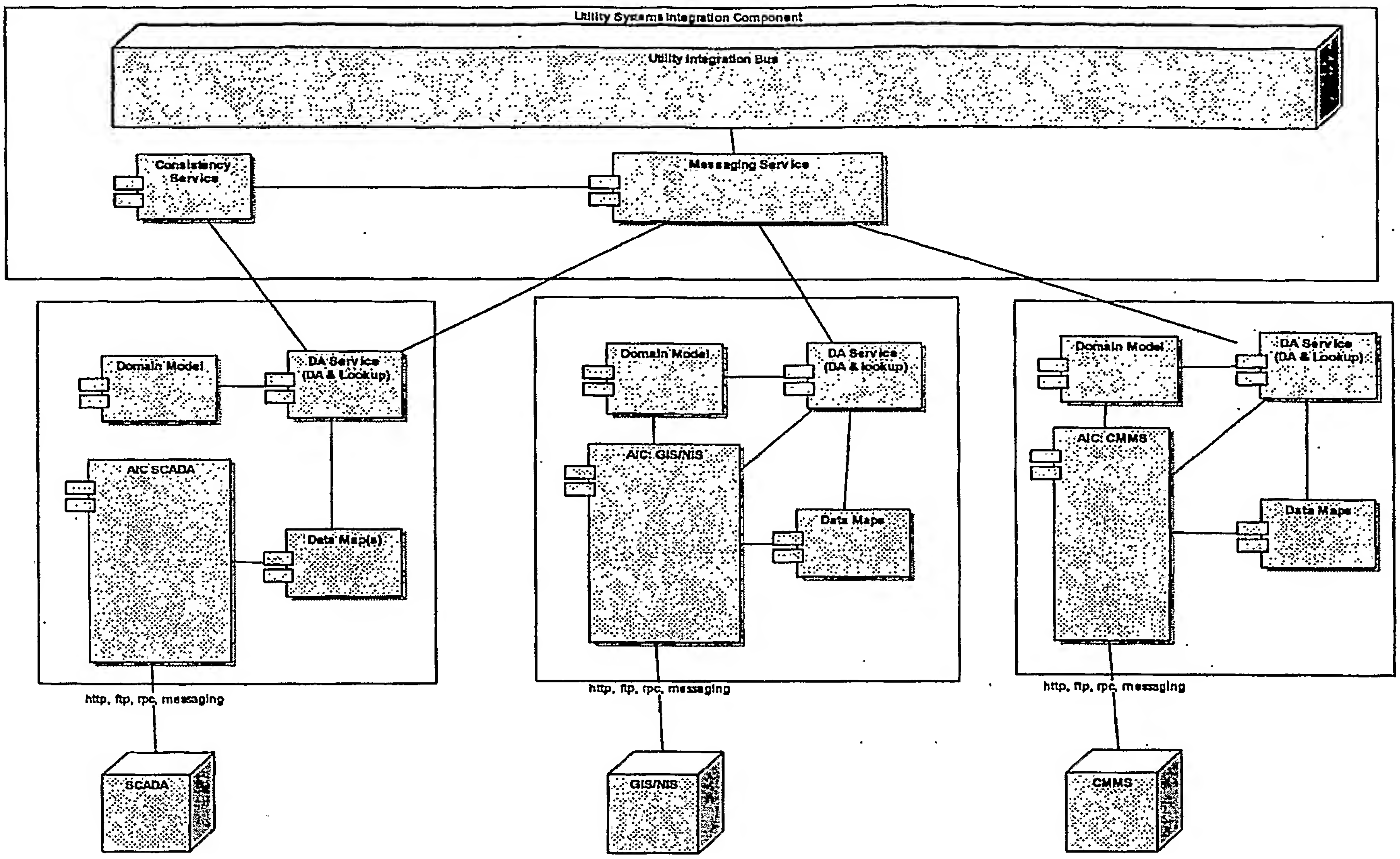
10.2.1 Description

In contrast to a "centralized" solution, it is also possible to extend the functionality of the application adapters with a local lookup and data access services. Basically, the data mapping from a local format to a common CIM format, as well as the mapping of a *local object* to a common address space must be handled by each application adapter. The application adapters connect to a common messaging bus.

Client services, such as lookup and data access, and internal services, such as consistency and synchronization, communicate with the application adapters through the messaging bus. If a client application does a lookup, the common service basically "asks" all connected application adapters if they can provide a match to the requested query. If yes, the data access service asks all application adapters to execute a data access request. The data access service assembles from the individual responses a response towards the client application.

Services like consistency and synchronization are monitoring messages, and if a message is identified which will require, e.g., a consistency check, the consistency service itself executes the necessary actions.

Before a target application can participate in the messaging, it must translate its address space to the address space given by the USI domain model (CIM instance naming).



10.2.2 Discussion

- Main advantage of a decentralized solution is the effective distribution of processing. The services issue requests through messages, and the connected application adapters can pick up the requests and perform certain actions (also in parallel).
- Additional services may be hooked up more easily into the concept than in the centralized solution by listening to messages which are exchanged, and responding to message classes or instances.

- A decentralized solution only works on a given global addressing scheme, since queries can only be resolved towards a common address space. It means that *global objects* need to be assigned a unique identifier at configuration time.

- Each application adapter implements the functionality lookup (for the application *local objects* to the global address space), and data access. Therefore, EAI components must be licensed for each application adapter.

11 REFERENCES

[1] CRID30020-03-002 G0 Report

[2] CRID30020-03-P2-008 Requirement Specification Phase 2

[3] CRID30020-03-003 Project Plan USI

[4] CRID30020-03-010_P1 Architecture Document Phase 1

12 REVISION HISTORY

Rev.	Chapter	Description	Date Dept. / Name
1.0	All	Initial document authoring	06.08.2003, Claus Vetter
1.1	All	Reworked, added chapters on general concepts	14.08.2003, T. Werner
1.2	3, 4, 5	Reworked after review	23.9.03, T. Werner
1.3	All	Replacements, Deleted Appendix, smaller corrections	24.09.2003, Claus Vetter